Novel Optimization Metaheuristics and their Applications

A Dissertation submitted to the University of Hyderabad in partial fulfillment of the degree of

MASTER OF TECHNOLOGY

in

Information Technology

by

Y. Mahesh Kumar

Department of Computer and Information Sciences

School of Mathematics, Computer and Information Sciences

University of Hyderabad

(P.O.) Central University, Gachibowli

Hyderabad – 500 046

Andhra Pradesh

India

CERTIFICATE

This is to certify that the dissertation **“Novel Optimization Metaheuristics and their Applications”,** submitted by **Mr. Y. Mahesh Kumar** bearing Reg. No. **09MCMB39** in partial fulfillment of the requirements for the award of Master of Technology in **Information Technology** (**with specialization in Banking Technology and Information Security**) is a bonafide work carried out by him/her under my supervision and guidance.

The dissertation has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

Dr. V. Ravi
Signature of the Supervisor

Head of the Department

Dean of the School

DECLARATION

I **Y. Mahesh Kumar** hereby declare that this Dissertation entitled "**Novel Optimization Metaheuristics and their Applications**", submitted by me under the guidance and supervision of **Dr. V. Ravi, Associate Professor, IDRBT**, is a bonafide work. I also declare that it has not been submitted previously in part or in full to this University or other University or Institution for the award of any degree or diploma.

Date:

Name: **Y. Mahesh Kumar**

Signature of the Student

Regd. No. **09MCMB39**

Acknowledgements

I take this opportunity to thank everyone who helped me in doing the research during my M.Tech project.

Firstly, I thank the Almighty for making everything possible.

I would like to express my sincere gratitude to Dr. V. Ravi who supervised the project. I am very thankful for his continuous support not only as a guide but also as a mentor helping me in exploring new research areas and guiding me towards the right path. This work would not have been possible without the broad vision and assistance of Dr. V. Ravi.

I thank Mr. B. Sambamurthy, Director, IDRBT, Dr. Arun Agarwal, Dean, MCIS, Dr. C. Raghavendra Rao, Head of the Department (DCIS), University of Hyderabad for extending their cooperation.

I thank Dr. Mahil Carr, M.Tech-IT Coordinator, IDRBT for his kind support during my project work. I thank all the faculty of IDRBT and DCIS, University of Hyderabad for all the courses they taught me which helped me in completion of my project.

I want to express my deep faith and love to my parents for their everlasting support.

I would like to thank Dr. Abdul, who supported me a lot with his kind advices and moral support during my project. I want to thank all my friends and colleagues for their necessary support and encouragement throughout my M.Tech course.

Communicated Research Papers

1. Y. MaheshKumar, V. Ravi and A. Abraham: A PSO-TA Hybrid Algorithm for Unconstrained Optimization (**selected in the special issues of NABIC 09**).

2. Ravidutta Chowdary, V. Ravi and Y. Mahesh Kumar (2010): A Hybrid Harmony Search and Modified Great Deluge Algorithm for Unconstrained Optimization, ***International Journal of Computational Intelligence Research***, Vol.6 Issue 4, pp.755-761.

3. Y. MaheshKumar and V. Ravi: A Hybrid Filter-Wrapper Method for Feature Subset Selection (paper to be communicated).

4. Y.Mahesh Kumar and V. Ravi: A Hybrid Harmony Search and Threshold Accepting Algorithm for Unconstrained Optimization (paper to be communicated).

Abstract

The quench for optimization does not dry at all in any realm at any time. Many methods were proposed and being proposed to provide optimality. There is no realm without any need for optimization. This desperate need led to rapid research progress in the field of optimization. Metaheuristics which were developed by inspiring from the phenomenon happened in the nature provide efficient solutions for optimization problems. The greatest advantage of these Metaheuristics is that these methodologies can be applicable to any other application where optimization is needed. Feature subset selection using the help of Metaheuristics drives the process of selecting the optimal features faster with high efficiency. Feature subset selection is one of the most important research realms in data mining which is used at data preprocessing. Feature Subset Selection is a procedure followed for dimensionality reduction of the dataset without losing the essence of the data by finding the features which have less impact on target variable and ignoring them for building robust machine learning models, one of the most challenging jobs in data mining.

Firstly, we developed a novel metaheuristic using Particle Swarm Optimization (PSO) and Threshold Accepting (TA) method. The searching procedure initially starts with PSO and then the worst solution for that iteration was fine tuned using TA probabilistically. In our approach TA is made tightly coupled with PSO and is called every time at the end of PSO's iteration. This approach of biased sampling of TA with PSO provided good results when we tested on benchmark problems. The efficacy of our proposed model is tested by comparing its performance on several benchmark problems of different dimensions with traditional PSO. Our proposed model outmatched the traditional PSO in terms of accuracies, functional evaluations, standard deviations of accuracies, success rates.

Then we move on developing other novel Metaheuristics using Harmony Search, Modified Great Deluge Algorithm (MGDA), Threshold Accepting (TA) methods. We developed two hybrid models: Modified Harmony Search (MHS) + TA, MHS+MGDA. The efficacies of the developed models were tested on benchmark problem where our methodologies outperformed the traditional harmony search in terms of accuracies and functional evaluations. When we compare the performance of MHS+MGDA with

MHS+TA, MHS+TA provided better results. The rigorous testing of several approaches to provide a great insight into several metaheuristic methods during my research and encouraged me to develop an application using metaheuristic.

We developed a metaheuristic based application for feature subset selection. A novel hybrid model for feature subset selection which uses the efficacies of both filter and wrapper methods were proposed. The wrappers are optimization based wrappers where an optimization metaheuristic wrapped around the classifier. Here we used binary PSO as an optimization algorithm with Multi Layer Perceptron (MLP), Threshold accepting trained Logistic Regression (TALR) as classifiers and t-statistic, Mutual Information as filters. The proposed approach was tested on 7 most popular binary classification datasets of different number of features and patterns. Our proposed model provided efficient classification accuracy with almost half of the features.

Table of Contents

1

Introduction

1.1 Overview of Optimization and Metaheuristics

Optimization is as ageless as time and is omnipresent, existing in every realm. It is a process of attempting to find the best possible solution out of all possible solutions. The task of optimization is to model the given problem into an evaluation function which represents the quality of a given solution and then employ a search method to optimize (minimize or maximize based on the given problem) the objective function (Edmund and Graham, 2005). When we are searching for optimal solution in the given search space we will come across two types of optimal solutions; local optimum and global optimum. A local optimal is a point in a search space where all the neighboring solutions are better than the current solution and the global optimum is a point in the search space where all other points in the search space are worse than or equal to the current solution. Many search methodologies were proposed to find the optimal solutions which include exhaustive search, heuristics, Metaheuristics, evolutionary methods, etc.

Heuristics generally referred to as trial-and-error methods. A heuristic is a method which seeks good solutions at a reasonable computation cost without being able to guarantee optimality, and possibly not feasibility and it is difficult to state how close to optimality the solution is (Reeves, 1996). Heuristic methods are very valuable most of the time but the results obtained are not optimality guaranteed. There were many variants of heuristic methods like local search heuristics, constructive heuristics, hyper heuristics, etc.

Metaheuristics refers to set of concepts that can be used to define heuristic methods such that they can be applied to a wide set of different problems. They can be seen as a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to a specific problem. It was first named by Fred Glover in 1997. He referred metaheuristic as a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality. The heuristics guided by such a meta-strategy may be high level procedures or may embody nothing more than a description of available moves for transforming one solution into another, together with an associated evaluation rule. At present Metaheuristics is one of the important research areas in searching methodologies. In literature the term Metaheuristics refer to broad collection of relatively sophisticated heuristic methods like Particle swarm optimization, Differential evolution, Tabu search, Genetic algorithms, etc., . The term 'Metaheuristic' was sometimes interchanged as 'modern heuristics' in the literature.

1.2 Overview of Feature Subset Selection

Feature Subset Selection is a procedure followed for dimensionality reduction of the dataset without losing the essence of the data by finding the features which have less impact on target variable and ignoring them for building robust machine learning models, one of the most challenging jobs in data mining. Several methods were proposed for feature subset selection like filter methods, wrapper methods, hybrid of filter and wrapper methods. Each method is significant in its own way and every method has its own drawbacks. Filters are one of the feature selection methods that consider individual feature relevance to the target variable. Wrapper approach is an efficient feature selection method that provides an optimal feature subset that efficiently solves the classification or prediction tasks. There are two types of wrappers: wrappers developed using a classifier or function approximate and wrappers with optimization algorithm wrapped around which gives the feature subsets as solutions to classifier.

1.3 Motivation

Developing new methodologies to solve optimization problems is gaining increasingly important these days. As the applicability of optimization is huge there is big need in developing an efficient methodology to provide the best possible solution. The process of developing new methodologies is a challenging and highly enthusiastic which encouraged me very much in taking up this project. The Metaheuristics which were already present in the literature mostly developed by inspiring from the nature. If we consider particle swarm optimization on which I worked, it was based on how a flock of birds or fish move towards the destination in an optimal path. Consider ant colony optimization, the way in which ants search and transport their food inspires to develop the algorithm.

The other problem which I worked is about feature subset selection. Feature subset selection in a broad view can be considered as an optimization problem where we try to find out an optimal feature subset which gives high classification accuracy. Feature subset selection is about finding the features which are more important and well sufficient for developing any classification or prediction model which is a spine-tingling job.

1.4 Objectives

The main objective of my research is:

- Analyzing different metaheuristic methodologies proposed for optimization.
- Developing novel Metaheuristics which are more efficient than the ones proposed in the literature.
- Testing the efficacy of developed novel Metaheuristics on several benchmark problems and also comparing with other approaches.
- Analyzing different methods developed for feature subset selection.
- Developing a novel methodology for feature subset selection and test its efficacy on several benchmark datasets.

1.5 Organization of the thesis

Chapter 2 starts with describing particle swarm optimization algorithm and then threshold accepting algorithm is explained. It was followed by a review of many hybrid methodologies developed to solve optimization problems. A novel optimization metaheuristic using Particle swarm Optimization and Threshold accepting algorithm was described and the results obtained on different benchmark problems are discussed. Chapter 3 describes Harmony Search, Modified Great Deluge Algorithm and then several hybrid Metaheuristics proposed in the literature are reviewed. Two novel Metaheuristics developed using harmony search and threshold accepting algorithm and Modified Great Deluge Algorithm were described and the results of the model on different benchmark problems are discussed and then conclusions are drawn. Chapter 4 describes different feature subset selection methods and different hybrid methodologies proposed to find optimal feature subsets are discussed. The methodologies used in the model are briefly described. A novel feature subset selection method using metaheuristic was developed and its efficacy on different datasets are discussed. Overall conclusions of the work were presented in chapter 5.

2

A Particle Swarm Optimization and Threshold Accepting Hybrid Metaheuristic

2.1 Particle Swarm Optimization Algorithm

Inspired by the choreography of swarm of birds, a novel population based metaheuristic named PSO was developed by Kennedy and Eberhart (1995) which is very efficient in solving optimization problems. It was developed based on the idea of how a swarm of birds or fish move in search of their food. A swarm is a set of disorganized moving individuals that tend to cluster where each individual is moving in random direction. The choreography of these swarms inspired Kennedy and Eberhart (1995) to formulate it into a powerful meta-heuristic.

PSO is very simple to implement and efficacious. It has very few parameters to tweak and that feature makes it simpler and also by the absence of greediness in the algorithmic design makes it faster. PSO progresses towards the solution by mutual sharing of knowledge of every particle collectively.

PSO consists of swarm of particles where each particle has its own position and velocity. Each particle is initialized randomly at the beginning of the algorithm and the heuristics update the position and velocity in latter stages of algorithm. The change in position of the particle is influenced by particle's own experience and that of best particle in the swarm i.e. the particle that provide better optimal solution than any other particle in the swarm . Each particle is evaluated using fitness function that indicates the closeness to optimal solution. Every particle has a memory variable that stores the best position obtained by the particle so far and is termed as *pibest* where *pibest* is the best position for ith particle. The particle that provides best fitness value in the swarm is stored as *gbest*. At each iteration the position of the particle in each dimension is updated by adding velocity to its position making it move towards *gbest* and *pibest*.

The pseudo code of PSO for global minimization problems is as follows:

Let P ={p1,p2,p3,…..pn} be set of particles where each pi is of d dimensions; pid={pi1,pi2....pid} Each particle has its own individual velocities vi i.e., V={ v1, v2, v3,…… vn}

Start

Initialize positions and velocities randomly of permissible range for each particle.

While convergence criteria is not met

DO

For each particle

  Calculate the fitness value

  If the fitness value of the particle is less than pibest (old) then

    Update pibest to present value

  End if

End

Update gbest with the particle that provides best fitness value of all the particles in the swarm

For each particle

  For each dimension

    *vid= w*vid(old) +c1*rand*(pibestd-pid)+c2*rand*(gibestd-pid) // update particles velocity*

    pid = pid (old)+vid   // updating particles position

   end for

end for

END

Where c1, c2 are acceleration coefficients, rand: random number between 0, 1 and w is the inertia weight.

The quantities *pibestd* and *gibestd* share the knowledge of particles previous experience and experience of whole swarm. The way in which birds in a flock move towards their food by considering their experience and taking the help of their neighbors to move further, each particle in PSO like a bird in the flock moves towards the optimal solution.

2.2 Threshold Accepting Algorithm

Threshold Accepting algorithm was proposed by Dueck and Sheur (1990). It is a point based search technique. It is a variation of Simulated Annealing (SA) algorithm while in SA new solutions are accepted probabilistically and in TA they are accepted based on a deterministic criterion. In TA any new solution that is not much worse than the previous solution is accepted.

The pseudo code of TA is as follows:

Initialize the solution and set global iteration counter itr=0, old=99999, thresh=2

```
fi ← fitness value of initial solution
while itr<gitr    // gitr is the number of global iterations
DO
    itr ← itr+1
    ii ← 0            // ii - inner iteration value
    while ii < limit or del1 > thresh
    DO
        ii ← ii+ 1
        Generate a candidate solution vector using the following equation
            candidate solution = old solution+(max-min)*(2*u-1)pindex
```

fj ← fitness value for the candidate solution

del1 ← fi – fj

END

If del1 < thresh , set fi = fj

If thresh < thrtol , set del2 = (new – old) / old

Report current solution as the optimal one if abs (del2) < acc and exit if itr < gitr

Else

old ← new

*thresh = thresh * (1-eps)*

END

TA is applied on a single solution. The algorithm runs for '*gitr*' number of global iterations and for every inner iteration ii a candidate solution is generated. The fitness value is calculated for each candidate solution and the solutions that are not much worse than the previous one are selected for exploring. The algorithm terminates when the difference between previous and present objective function values is very small as determined by the parameter acc which is set to 10^{-6} to obtain highly accurate solution. The parameter thresh is used to determine the acceptance of candidate solution and is generally set to 2 at the beginning and is gradually decreased in a geometric progression based on an epsilon value that is generally set to 0.01. *limit* is the number of inner iterations. *max*, *min* are the boundaries of the decision variables and *pindex* is generally an odd integer between 3 and 33 and is used to generate a value that is added to the old solution to generate a neighborhood solution.

2.3 Overview of Hybrid Metaheuristics developed for optimization problems

Many researchers use Metaheuristics as an efficient way of solving optimization problems. Differential Evolution, a global optimization meta-heuristic, proposed by Storn and Price (1997), is a simple and efficient algorithm. Threshold Accepting, proposed by Dueck and Scheur (1990), is a local search technique that works on individual solution to search for the optimal state. Global search techniques like PSO, DE, ACO, etc., are highly efficient in searching for ‘promising regions’, while exploitation could be very well done through local search techniques like Nelder-Mead Simplex Search (NMSS), Threshold Accepting (TA).

The process of hybridization of the existing Metaheuristics is followed by researchers over the last 15 years. The hybrid optimization algorithms benefit from the advantages of the component metaheuristic algorithms. To start with, Ravi et. al. (1997) hybridized Non-Equilibrium Simulated Annealing (NESA) with a simplex like heuristic to develop a new algorithm called Improved Non-Equilibrium Simulated Annealing (INESA). This is one of the earliest hybrid algorithm proposed in the literature. In this paper, they improved Non-Equilibrium Simulated Annealing (NESA) by taking the solutions at regular intervals of the progress of NESA and then combining them with the best solutions obtained before the termination of NESA part of algorithm. At this stage they applied a simplex-like heuristic to obtain the global optimum. Chauhan and Ravi (2010) hybridized Differential Evolution (DE) with Threshold Accepting (TA) that takes the advantage of efficient exploration of DE and exploitation of TA. They reported spectacular reduction function evaluations when tested on test problems. Schmidt and Thierauf (2005) hybridized TA and DE where TA is first applied to all solutions of solution space and the resultant was passed to DE to move towards global optimal solution. Bhat et. al. (2006) hybridized DE by employing reflection property of the simplex method for fast convergence to global optima. Trafalis et. al. (2002) hybridized 3 heuristics namely scatter search, GA and TS. They introduced the notion of memory to explore the solution space more extensively. It used the scatter search by combining the concepts of trajectory and clustering methods. The later stages of the algorithm combined the characteristics of TS and GA to test the status of new solutions and to direct the search towards global optima. Srinivas and Rangaiah (2007) developed a hybrid using DE by using tabu lists for solving global optimization problems. Chelouah et al. (2003) hybridized GA and NMSS search method. They used GA to do detect promising regions where we can find the optimal solution and used NMSS for Intensification i.e., to locally search for global optimum in this promising region.

Many hybrid Metaheuristics were proposed in the literature using PSO. Deep and Bansal, (2009) hybridized PSO with Quadratic Approximation where the whole swarm is split into two sub-swarms and PSO is applied on one sub-swarm and QA on other, ensuring that sub-swarms are updated using the global best particle of whole swarm. Milie et al. (2009) improved PSO by including cross over operator to maintain diversity in the PSO. Li et al. (2009) hybridized PSO with improved Genetic algorithm (IGA) where the worst solutions in every iteration particles providing worst solutions are passed to IGA and remaining are modified using PSO. Shelokar et al. (2006) hybridized PSO with Ant Colony Optimization (ACO) for solving highly non-convex optimization problems. Here firstly PSO explores the search space thoroughly and when reaching the promising area the solutions are passed to ACO that does the exploitation part by performing local search. Here the concepts of diversification and intensification are used which were first introduced by Glover (1989). Diversification refers to finding the promising regions that has most probable solution by considering entire search space. Intensification is to locally search for global optimum in the promising regions. Jiang et al. (2007) proposed an Improved PSO by dividing the whole swarm into several sub-swarms and PSO is applied on each sub-swarm individually and at periodic stages of evolution the population is shuffled to ensure information sharing. Fan and Zahara (2007) hybridized Nelder-Mead Simplex Search (NMSS) with PSO where at every stage of evolution the particle providing elite solutions are passed to NMSS and the worst one are passed to PSO. Kulakarni and Moorthy, (2007) proposed an estimation of distribution improved PSO where restricted the particles to best solutions by probabilistic modeling of archive of best solutions. Zhang and Xie, (2003) hybridized PSO with Differential evolution operator (DEPSO) which provide the bell-shaped mutations with consensus on the population diversity along with the evolution, while keeps the self-organized particle swarm dynamics. Ben and Li, (2009) developed a hybrid meta-heuristic using PSO and DE where they divided the entire swarm into two in which PSO and DE are applied on two sub swarms and move towards optimal solution ensuring that they both share same global best particle. Salhi et. al. (2004) hybridized SA, TS and a descent method. Their main focus was on optimizing functions with several minima. The technique along with finding the global minima, also finds equal global minima, which are almost as good as the best one. They used SA to determine the promising regions. On that they construct tabu regions applying TS. Finally, a descent method is applied to point out all global optima. Chelouah et. al.

(2005) hybridized TS and NMSS. Arun and Ravi, (2009) hybridized ACO with NM Simplex Method.

2.4 A Particle Swarm Optimization Algorithm and Threshold Accepting based hybrid Metaheuristic

A new hybrid algorithm was proposed by tightly coupling PSO with TA. In this approach the searching procedure initially starts with PSO and then the worst solution for that iteration was fine tuned using TA probabilistically. After the swarm initialization, the particles are updated as in traditional PSO and at the end of every iteration we generate a random number between 0 and 1. If the random number ≥ 0.9, then we invoke TA to update the worst solution, consequently we were invoking TA probabilistically. The worst solution is the solution or particle that gives worst fitness value of all the particles. Here we set PSO as the main optimizer and use TA to tweak the worst solution. This is made with an idea that the worst solution that fails to provide a better fitness value took the help of TA to move towards the optimal solution. TA, which is a powerful local search technique ushers the particle in moving towards global optimal solution. Here we are not using TA for whole particles to provide the global optimal solution but applied finely that helps in reenergizing the particle towards optimal solution and then the updated particle is made to rejoin the swarm for the next round of PSO. In this way the process is repeated until convergence criteria is met.

The step wise description of PSOTA hybrid for global minimization problems is as follows:

1. *Initialize the particle swarm*

   Here every particle is initialized randomly within the range of the decision variables in the problem and initialize the velocities of the particles within the range [0, 1]. The acceleration constants c1, c2 are set to 2 and the inertia is set to 1. Coming to TA parameters epsilon '*eps*' is set to 0.01, acc set to 0.0000012, *thrtol* to 0.000001, *pindex* is set to 29.

2. *While the convergence criteria is not met*
3. *DO*

4. *For each particle*

4.1 Calculate the fitness value by computing objective function for the particle.

4.2 If the fitness value of the particle is less than pibest (old) then

3.2.1 Update pibest to present value

End if

End for

5. *Update gbest with the particle that provides best fitness value of all the particles in the swarm*

6. *For each particle*

5.1 For each dimension

*vid= w*vid(old) +c1*rand*(pibestd - pid)+c2*rand*(gibestd - pid) // update particle's velocity*

pid = pid (old)+vid // updating particle's position

end for

end for

7. *if (rand() ≥ 0.9) then*

Calculate Threshold Accepting algorithm (TA) for the worst particle.

End if

8. *Replace the worst particle in the swarm by the solution given by TA*

9. *If a convergence criterion was satisfied the PSO is called and algorithm steps from 3 are repeated.*

10. *Else algorithm exits providing the obtained optimal solution*

11. *END*

We invoke TA whenever the random number generated ≥ 0.9 i.e., with 10% probability. This sort of biased sampling helps in reducing the overhead of TA on PSO. Here every particle is of '*d*' dimensions. Here the values of limit and *gitr* are set differently for lower and higher dimensional problems which are specified in the experimental settings. The sequence of event flow is presented pictorially in the form of flow chart denoted as Figure 1.

By incorporating the flavor of TA the performance of PSO is considerably enhanced. The main intention of invoking TA probabilistically is to enhance the traditional PSO without having any overload of TA. This approach pushes up the particle that lags behind other particles in that iteration and rejoins the swarm with more refined and continues to search the global optimal solution. Here we use TA to update only the worst particle because we find that at every iteration worst particle is the particle that gains least knowledge from the other particles in the swarm and thus by invoking TA will help to reenergize the particle such that it will certainly move towards near optimal solution in next iteration. The use of TA probabilistically will help us in having less functional evaluations and also ensures that the robustness of PSO is preserved. This approach really helped in reduction of functional evaluations without any compromise in the accuracy which was described next in the paper. The success rate of our hybrid is higher when compared to normal PSO. Our proposed approach had utilized the best features of both PSO and TA in a very balanced manner. The schematic representation of our proposed approach is presented in Figure 2.

2.5 Results and Discussions

The proposed algorithm is tested on 34 unconstrained optimization problems. Most of the problems are taken from Ali et al. (2005) and some from web. It is run for 30 simulations for different random seeds. The average, standard deviations of the objective functions over 30 simulations are computed. The experimental settings for the proposed hybrid are presented in Table 1.

The algorithms PSO, PSOTA are implemented in ANSI C using C-Free 4.0. The tests are made on a system with Intel Pentium 4 processor with 2.39 GHz clock speed and 1.99GB of RAM. Both PSO and PSOTA hybrid are run on 34 unconstrained optimization problems (26 lower dimensional and 8 higher dimensional). The performance of PSO and PSOTA is presented in Tables 3, 4, 5 and 6. The results indicate the performance of PSOTA in terms of accuracy and consistency. The proposed algorithm provides high accuracy in some cases and at least as same accuracy as that of PSO with less functional evaluations. The use of TA to refine the particle helped us in reducing the functional evaluations and also improved accuracy.

Table 3 presents comparativeness of PSO and PSOTA in terms of accuracy and functional evaluations for lower dimensional problems. The result shows that for most of the specified optimization problems, PSOTA provides efficient result with less functional evaluations. For some problems like Camelback6, Becker and Lago, kowalik, Levy and Montalov1, Miele, Axis Parallel Hyper Ellipsoidal, Shekel 5, 7, 10 problems the hybrid outperformed PSO by providing better results with very less functional evaluations. Table 4 presents the performance of PSO and PSOTA in terms of accuracy and functional evaluations in case of higher dimensional problems. In the first column number@dimesion notation is used, for example, 1@30d means the function of serial number 1 in Table 2 of 30 dimensions. Here dimension indicates the number of decision variables. The results show the robustness of the proposed hybrid. The proposed hybrid provides better solutions with less functional evaluations. As regards higher dimensional problems, RosenBrock (20dimensions), Sphere (30 dimensions), Levy and Montalov2 (30 dimensions) and Zakharov (50 dimensions) the proposed hybrid out performed PSO in terms of functional evaluations. Table 5 and 6 compares PSO and PSOTA in terms of Success Rate and Standard Deviation. Success Rate is considered based on the results obtained in the 30 simulations. Here we considered the resultant value as successful if the obtained objective function value is within 10-6 difference of the reported value. So the success rate is defined as number of successful results / total number of simulations *100. Table 5 indicates that the success rate of PSOTA is better and consistent than that of PSO. The consistency of PSOTA is its main asset. Standard deviations of PSOTA and PSO are also tabulated in which the proposed hybrid outperformed PSO for many functions. Table 6 presents success rate and standard deviation values for higher dimensional problems by PSO and PSOTA. The success rate of the hybrid is better than that of PSO and when it comes to standard deviation values our hybrid outperformed PSO in all functions.

Graphical presentation of results:

The performance of PSO and PSOTA is depicted in Figures 3-10, where the dark line denotes our proposed PSOTA hybrid and dotted line denotes PSO. The optimal objective function values obtained by both algorithms for lower and higher dimensional problems are depicted in Figures 3 and 4 respectively. The average functional evaluations consumed by PSO and PSOTA for lower and higher dimensions are depicted in Figures 5 and 6 respectively. For the sake of convenience, we depicted the function evaluations consumed in the case of Shekel 5, 7 and 10 functions in Figure 6 because there both algorithms consumed very high number of function evaluations. The success rate obtained by PSO and PSOTA for both lower and higher dimensional problems is depicted in Figures 7 and 8 respectively, whereas the standard deviation values obtained by PSO and PSOTA for lower and higher dimensional problems are depicted in Figures 9 and 10 respectively. Evidently, all the above mentioned plots demonstrate the superiority of PSOTA over PSO not only in success rate but also in function evaluations.

2.6 Conclusions

The proposed hybrid meta-heuristic is compared with traditional PSO by testing it on set of 34 unconstrained problems. The results show that our hybrid provides consistent and efficient fitness values with less functional evaluations. The process of invoking TA at the end of each iteration probabilistically helped PSO approach the optimal solution faster. By using TA we are enabling PSO to do local search that it never does as per original algorithm. This process really helps it improvise PSO and also by biased sampling aspect of invocation of TA we are not increasing the complexity of the algorithm and also reducing functional evaluations taken to solve the objective function which were presented in the results section. Here TA acts like a catalyst that speed up PSO in moving towards global optimal solution by enhancing the performance of worst particle and also increasing its experience by making it do local search. Success Rate and standard deviation of PSO and our hybrid are compared in several graphs which demonstrate the robustness of our algorithm. Based on numerical results it is concluded that our proposed hybrid is more promising in providing global optimal solutions.

3

Hybrid Methodologies using Harmony Search

3.1 Harmony Search

Harmony search, proposed by Geem et al. (2001), is among the most recent meta-heuristic algorithms that found applications in science and engineering realms. Figure 12 describes harmony search process.

The novelty in the algorithm lies in the fact that it is analogous to the improvisation technique of musicians. The algorithm in brief applies the idea of building an experience and then produces the best result that can be obtained from this experience. The analogy is such that a musical instrument represents a decision variable, its pitch range represents the value range, solution vector is represented by the harmony and with thorough iterations (analogous to practice) we improve the objective function which is represented by the aesthetics. Given the random nature of the technique, it is highly likely that it will escape the local optima. An added benefit its simplistic nature provides is the fact that it performs very little operation on each prospective solution thereby substantially reducing program execution time. But, one of the major issues with Harmony Search is that for prolonged periods of time (in terms of iterations), during the execution of the program, its solution remains unchanged, especially during the final stages. As a result of which several iterations are performed with no genuine improvement to the solution.

3.2 Modified Great Deluge Algorithm

Dueck et al. (2004) proposed Great Deluge Algorithm, a point-based search is an extension of the threshold accepting algorithm. It mimics the problem of obtaining the highest point in a piece of land by evading the water as it rains continuously. As the water level increases, the algorithm tries to avoid the wet land and eventually land at some high point. Although it may be considered similar to the SA algorithm it differs by virtue of its analogy to the water lever idea instead of the temperature parameter. An improvement to this algorithm was proposed by Ravi et al., (2004) wherein a new initialization scheme and neighborhood search scheme were designed and included in GDA, giving rise to MGDA. He inferred that MGDA may be considered an alternative to other algorithms like Simulated Annealing, Improved Non Equilibrium Simulated Annealing and Ant Colony Optimization. The effectiveness of MGDA was tested in the field of reliability optimization problems. It was observed that like other metaheuristic algorithms the choice of the powerful search scheme and updating of controlling parameters is critical to the performance, MGDA.

The distinctive improvement that this modification makes to GDA is the fact that it assigns the objective function value of the first solution to the parameter LEVEL thereby making the algorithm more robust and requiring less problem based parameter setting. Another modification is the introduction of the parameter limit in order to increase the speed of convergence. Thus the speed and accuracy of the algorithm are substantially improved.

3.3 Literature review of Hybrid Metaheuristics using Harmony Search

The approach of hybridizing several Metaheuristics, takes advantage of achieving both efficient "exploration" and "exploitation" in the process. Several earlier works followed that approach. Several hybrid Metaheuristics developed for optimization problems were presented in the previous chapter and here we present some of the hybrids developed using harmony search were presented.

Many hybrid methodologies using harmony search have been proposed in the literature. Fesanghary et al. (2008) developed a hybrid metaheuristic using harmony search and sequential Quadratic Programming. Here after employing harmony search Sequential quadratic programming is applied on each solution to perform local search. The solution which provides better objective function value than other solutions is considered as the final accepted solution. Two modified HS methods to deal with the uni-modal and multi-modal optimization problems have been proposed by Gao et al, (2009). The first modified HS method is based on the fusion of the HS and Differential Evolution (DE) technique, namely, HS-DE. The DE is employed here to optimize the members of the HS memory. The second modified HS method utilizes a novel HS memory management approach, and it targets at handling the multi-modal problems. A heuristic particle swarm ant colony optimization (HPSACO) is presented for optimum design of trusses by Kaveh and Talatahari, (2009). The algorithm is based on the particle swarm optimizer with passive congregation (PSOPC), ant colony optimization and harmony search scheme. HPSACO applies PSOPC for global optimization and the ant colony approach is used to update positions of particles to attain the feasible solution space. HPSACO handles the problem-specific constraints using a fly-back mechanism, and harmony search scheme deals with variable constraints. A novel hybrid metaheuristic using harmony and PSO was developed by Li and Li (2007). In this approach they induced harmony search inside the PSO such that before updating the position and velocity the solutions are passed as initial vector to harmony search such that a new solution generated every time is compared with worst solution and updated. Pan et al. (2011) proposed a local-best harmony search (HS) algorithm with dynamic sub-harmony memories (HM), namely DLHS algorithm, is proposed to minimize the total weighted earliness and tardiness penalties for a lot-streaming flow shop scheduling problem with equal-size sub-lots.

3.4 Modified Harmony Search

Two modifications were proposed for Harmony Search algorithm.

-- The value of *hmcr* (probability of selection of solution vector components from the Harmony Memory) is kept dynamically increasing from 0 to 1 during the execution of the program. This is done such that in the initial stages the new vector generation is more out of random initialization rather than selection from memory. This allows greater scope for the harmony memory to improve. As the number of iterations pass the value of *hmcr* tends to one.

-- The HS algorithm is terminated when the difference between the best and the worst solution in the harmony memory is found to be less than some predefined constant. The HS algorithm with these modifications will hereon be referred to as the Modified Harmony Search (MHS).

3.5 Hybrid Metaheuristic using Modified Harmony Search and Modified Great Deluge Algorithm

While in most of the cases the unnecessary iterations of HS may be averted by choosing a fixed number of iterations of the algorithm within which the algorithm will converge to the solution with certain amount of accuracy, it has its share of drawbacks. An incorrect judgment on the part of fixing of the iterations can either lead to an early exit of the algorithm without reaching the desired result or delayed termination for the same amount of accuracy because of undesired number of excessive executions.

On the other hand MDGA is one such algorithm that has been proved to perform satisfactorily in various applications (Ravi and Pramodh, 2010). Thus, to obviate the excessive and unproductive iterations of MHS, we invoke MGDA after terminating MHS prematurely. However, MGDA, being a point-based search algorithm, is invoked to work on the best solution found by MHS.

We thus, propose an algorithm that will initiate the search for the optimal solution using the Harmony search, reach a tentative solution and then improve that using the MGDA.

Step by Step procedure followed by the MHS+MGDA hybrid:

Phase 1:

1. Generate a set of hms number of random solutions randomly and initialize harmony memory with this set.

2. Create a solution new vector with components of the solutions selected from harmony memory with a probability of *hmcr* such that the components when selected from the harmony memory are chosen randomly from different solutions within the harmony memory. Note that as the number of iterations increase the value of *hmcr* is increased linearly.

3. Perform the pitch adjustment operation by altering the variables' value by delta with a probability par ('delta' value is used in case of discrete optimization problems).

4. On evaluating the objective function value of this vector, if it is found to be better than the worst solution in the memory then replace the worst solution with this vector.

5. Repeat this procedure (Steps 2 through 4) till the difference between the best and worst solutions within the harmony memory falls less than some predefined diff1 (a small value), or maximum number of iterations (predefined) is reached (whichever happens earlier). This step terminates MHS and the best solution in the memory is then the optimal obtained solution by MHS. Note that we kept the value of diff1 larger than usual in order to facilitate early termination of MHS and begin with the next phase.

At the end of Phase I, the best solution obtained by MHS is then passed on to MGDA as the initial feasible solution.

Phase II:

1. The value of the objective function of the solution obtained at the end of Phase 1 is assigned to the variable f0 . Let the initial value of some parameters be as follows: itr=0, old=9999, LEVEL= f0 , UP is a user-specified number in the range (0,1) that determines, in the metaphor of original GDA, by which amount one has to climb the hill in order to prevent from being drenched. Similarly, LEVEL is the quantity by which water level goes up and itrmax is the total number of global iterations.
2. Increment the global iteration counter: itr=itr+1.
3. The inner iteration essentially performs neighborhood search. To accommodate this following stochastic procedure is performed and compared with resulting in a neighboring solution to the original one. $x^c_i = x^o_i + (2*u - 1)p$ where u is a random number drown from uniform distribution in the range (0,1), p is a pre-specified odd integer and the superscripts c and o indicate the candidate solution and the old solution respectively.
4. Compute the value of the objective function for the candidate solution and store in fc.
5. If (fc<LEVEL) then

   fo=fc,

   xoi=xci,

   LEVEL = LEVEL – UP * (LEVEL-fc),

   new = fo and go to the next step. Else go to step 8.

6. If (itr < itrmax)

If (new – old) < diff2)

Report xci=1,2,…., n as the global optimal solution with fc as the global optimum.

Else go to step 7

Else

Report xci , i=1,2,…,n as the global optimal solution with fc as the global optimum.

The value of diff2 is chosen as 0.000001 to obtain high accuracy. This stochastic perturbation does not necessarily result in a feasible solution. Hence, this perturbation is performed several times until a feasible solution is obtained. If the number of such trials, say, limit, exceeds a large pre-specified number, then it is understood/assumed that the algorithm is unable to find a feasible solution and hence the algorithm is forced to stop and the old solution at this stage is reported as the optimal solution. This parameter value is, however, problem-dependent.

The algorithm is computationally efficient with a time complexity of O (*hms**d) (Big-Oh) where *hms* is the harmony memory size and d is the number of decision variables in the problem. The space complexity of our hybrid optimization algorithm is O (*hms**d).

3.6 Hybrid Metaheuristic using Modified Harmony Search and Threshold Accepting Algorithm

From the experience gained from PSOTA model which was explained in the previous chapter the efficacy of Threshold Accepting algorithm in exploiting the search space was acknowledged. When harmony search was made to run on the benchmark problems the main problem, I observed is that the method is efficient in reaching the near neighborhood of optimal solution but it cannot exploit further. After running the algorithm for certain number of iteration the algorithm which works splendidly at the beginning but as the iterations progresses it is generating constant results. This motivated me to employ Threshold Accepting algorithm on the solution to improve further to find the global optimal solution. The procedure followed for MHS+TA methodology is as follows:

1. Harmony Memory is initialized with *hms* number of solutions which were generated randomly. The dimensionality of each solution depends on the type of problem.

2. During every iteration, a new solution vector with components of the solutions selected from harmony memory.

For each dimension of the new solution corresponding value from the harmony memory is considered with a probability of *hmcr* such that the components when selected from the harmony memory are chosen randomly from different solutions within the harmony memory. Note that as the number of iterations increase the value of *hmcr* is increased linearly since we are employing Modified harmony search.

3. Perform the pitch adjustment operation by altering the variables' value by delta with a probability *par* ('delta' value is used in case of discrete optimization problems).

4. On evaluating the objective function value of this vector, if it is found to be better than the worst solution in the memory then replace the worst solution with this vector.

5. Repeat this procedure (Steps 2 through 4) till the difference between the best and worst solutions within the harmony memory falls less than some predefined *diff1* (a small value), or maximum number of iterations (predefined) is reached (whichever happens earlier). This step terminates MHS and the best solution in the memory is then the optimal obtained solution by MHS. The diff1 value is set to 0.0001 such that we are terminating the MHS early to enable TA's help on the solution provided by the MHS. The best solution provided by the MHS is taken as the initial solution for TA and we run TA to find the best objective function value. The schematic view of our approach is presented in Figure 11. The best solution generated by the MHS was denoted by 'B' in the Figure 11. Here 1, 2, 3 ... N represents 'N' solutions in the harmony memory.

3.7 Results and Discussions

The proposed hybrid model MHS+MGDA was tested on 24 benchmark problems and reported the corresponding function evaluations. Table 8 presents the best, worst and average results of 30 runs for each of the algorithms by changing the seed for every run. In order to compare the two algorithms under the same conditions, the parameter settings of MHS and MHS+MGDA have been kept identical. For the generation of random numbers, the seed was changed based on computer clock time for each of the simulations. The modifications made to HS were kept intact while evaluating its performance without MGDA. This was done to lay emphasis on the hybridization of the two algorithms rather than express improvements to the Harmony Search. The comparison thus is between the modified Harmony Search and modified Harmony Search + MGDA. The simulation for MHS was terminated when the difference of the best and worst solutions was found to be less than 0.000001.The function evaluation obtained is the average of all the 30 simulations.

It was seen that MHS performed much better than regular HS for most of the benchmark problems. Hence it remains consistent with our idea of using MHS as the Phase I algorithm. Out of the 24 functions tested we found that the number of function evaluations was found to be significantly less for all the functions except in the case of RosenBrock function where there was a nominal increase. In terms of accuracy (average taken over the 30 simulations), in 14 function the hybrid out performed MHS and in 7 of the other functions the hybrid was found to perform to the same order of accuracy as MHS. In the remaining 3 functions the difference was found to be nominal. When we mention accuracy it is with respect to the average solution obtained over the 30 simulations with different random seeds in order to demonstrate the effectiveness of the hybrid algorithm. Table 7 presents the parameter setting for each of the problems for both the algorithms. Here *hms* is the harmony memory size, *hmcr* and *par* are the probabilities for selection from the harmony memory and pitch adjustment respectively. The parameter *fw* refers to the frequency width for the pitch adjustment. The parameter diff1 refers to the accuracy and is used to prematurely terminate the harmony search before invoking MGDA and *diff2* is the user defined tolerance parameter in the stopping criterion of MGDA. The values of *hmcr* presented in the Table 7 are the initial values of *hmcr*. The value of *diff2* is kept constant at 0.000001. The value of *diff1* is varied between 0.01 and 0.000001 depending on the problem in order to decide the point of termination of MHS. On a generic basis it can be clearly seen that this hybrid of MHS+MGDA is sufficiently robust and efficient in terms of accuracy and function evaluations compared to MHS.

The proposed hybrid model using MHS and TA was tested on 20 different benchmark problems of different dimensions. The efficacy of our proposed model was analyzed by comparing with the results of MHS, MHS+MGDA. Except for scaffe2 function our proposed hybrid produced excellent results by outperforming MHS and MHS+MGDA in terms of accuracy and functional evaluations. In the Table 9 the bracket values beside the functions specifies the dimensional value considered and the functions without any brackets appended to them are function with standard dimensions as specified in their reference. The results reported in the Table 9 are the average results of 30 simulations. This is made because of removing the influence of random numbers generated during the running of algorithm on the benchmark functions.

3.8 Conclusions

A hybrid algorithm involving MHS and MGDA in tandem was proposed. The effectiveness of the hybrid was tested on 24 problems taken from literature, Based on the numerical experiments, we the proposed hybrid outperformed MHS in all but one problem in terms of function evaluations. As regards accuracy, the proposed hybrid outperformed MHS in 15 problems. In the remaining 7 problems it yielded marginally inferior solutions compared to MHS. The effectiveness of the proposed hybrid was tested on higher dimensional problems like RosenBrock and Zakharov with 20 variables each. The hybrid model produced better results even for higher dimensional problems. Hence, we can conclude that the proposed hybrid using MHS and MGDA can be used as an efficient and sound alternative to MHS in solving unconstrained optimization problems.

A novel hybrid metaheuristic using MHS and TA was proposed. By analyzing the numerical results the proposed hybrid outperformed the remaining models in terms of accuracy and also producing the results by consuming very less functional evaluations. The proposed hybrid can be used as a sound alternative to MHS in solving unconstrained optimization problems if we want a model that that provides both accuracies with functional evaluation values at highly acceptable level.

4

A Novel Filter-Wrapper Hybrid Method for Feature Subset Selection

4.1 Introduction

Classification problem is an unwieldy job in data mining and whatever model that we use to solve it. Firstly, we train the model train with given patterns, and after the model was trained sufficiently it is tested on patterns which are unseen by the model during training process to check its performance. Generally, the real world datasets are high dimensional in nature and large in size which makes the data mining process more tedious. Feature subset selection provides the solution to this problem. In any given dataset we have some features which may be redundant or irrelevant influence the target variable more compared to the rest. The features which have low impact on target variable are traced and ignored and the features which are more significant are selected to solve the data mining problems. Many methods are proposed in the literature to solve the problem of high dimensionality. Basically there are two kinds of feature selection approaches: *Filters and Wrappers.*

Filters are one of the feature selection methods that consider individual feature relevance to the target variable. These methods find out how much significant is the feature in classifying the patterns of the dataset. They rank each feature in accordance to its relevance with the target variable and from those we can select the top ranked features and ignore least ranked features as they are less significant. Filters are computationally efficient but not in providing accurate solution. This is because a feature may influence the target variable well individually but inter variable dependencies are not considered.

Wrapper approach is an efficient feature selection method that provides an optimal feature subset that efficiently solves the classification or prediction tasks. They contain a classifier or prediction equation wrapped up by a powerful Metaheuristics. Here the job of metaheuristic is to initially provide a random subset of features from the total set of features in the dataset to the classifier, which tries to predict the class of the pattern. The classification accuracy provided by the classifier influences the selection of feature subset for the next iteration. In this way an optimal feature subset is provided by wrapper which gives high classification accuracy. Thus optimization based wrappers provide high accuracy but requires exhaustive computations.

On the other hand in non-optimization based wrappers, a classifier or regression model determines the importance of features while being trained so that at the end of model building it outputs the ranks of the features also for e.g.: an MLP, General Regression Neural Networks, Probabilistic Neural Networks, all decision trees, Grouping Method of Data Handling networks, etc.

Many filter and wrapper feature subset selection methods have been developed in literature but the main problem is that very seldom any two given methods yield identical feature subset as the optimal one. The reasons are that the underlying theory for all these methods is different. Also the feature subset selection is influenced by the training data. Hence, there is a need to develop a feature subset selection method which is model independent, training set independent and hybridizes the properties of both filter and wrapper methods. This led to development of hybrid methodologies (Chandra et al., 2009). In Ensembling methods, rather than using a single classifier, many classifiers are used and the top best classifiers are selected and then the best features from those selected by the classifiers are chosen using frequency based approaches.

However the problem with ensemble feature subset selection systems is the choice of various models for the classification or regression task. Recently another trend of hybridizing wrappers and filters named wrapper –filter hybrid approaches caught the attention of researchers.

4.2 Review of Feature Subset Selection methods

Feature Subset Selection is one of the most challenging problem in data mining to solve this problem many approaches were proposed. Mutual Information, a filter method was proposed by (Battiti, 1994). It was based on information theory and probability. In addition to that many filter methods like ReliefF, Correlation based filter, INTERACT were proposed in literature (Noliea et al. 2007). Filters mostly focus on how an individual feature influences the target variable. A filter method to find out how an individual feature interacts with the other features (Zhao and Liu, 2007). It tries to find out the feature which when considered with other features produces efficient results.

A Naïve Baye's Breadth First Search (NB-BFS) wrapper was proposed by (Kohavi and John 1997), which provides an optimal feature subset that are efficient in finding the target variable. They also compared the performance of the wrapper method with several other filter methods by comparing their performance on several datasets. A new wrapper approach based on neural networks was proposed by (Kabir et al. 2010). In this approach they used correlation based information in selecting features and determining characteristics of neural networks. Many neural network architectures for feature selection in Sputter process were proposed by (Ravi et al, 2004). They used Principal Component Analysis (PCA), GRNN, and MLFF for feature selection and CART to generate rules. Out of 77 features PCA, GRNN and MLFF selected 19 features as the most prior features.

Many optimization based wrappers are developed where the efficacy of optimization algorithms is added to classifier to enhance its performance. A Genetic Algorithm (GA) based wrapper was proposed by (Pendharkar and Rodger, 2004). They used GA to fine tune the weights of MLP. MLP was used as the classifier to classify the patterns. Another model in which GA was used to select the hyper parameters of SVM and SVM acting as a classifier to that wrapper was proposed by (Chen and Hsiao, 2008). A novel Ant Colony optimization method based feature selection method was proposed by (Basiri et al, 2008) where they used their model to predict post synaptic activity in proteins and also compared their model with PSO based wrapper approach (Lin et al, 2008) proposed a optimization based wrapper method in which they used PSO for parameter determination and feature selection of the SVM and also compared their model with GA+SVM model.

Many ensemble methods were proposed to handle the feature selection problem. (Chandra et al. 2009) proposed used many hybrid intelligent techniques and ensemble them to predict the failure of dotcom companies. Here they used MLP,LR, SVM, CART, FR and in them top 3 best classifiers are selected and from those based on majority voting they selected best features from them. (Ravisankar et al. 2010) proposed many neural network -genetic programming based hybrid methods for feature selection and the efficacy of their proposed hybrid methods were tested on dotcom dataset where the hybrid methods produced efficient results. Many hybrid and ensemble based soft computing techniques in bankruptcy prediction was surveyed in (Verikas et al, 2010). In this paper they described many hybrid and ensemble techniques using GA, Rough Sets (RS), C4.5, Probabilistic Neural Networks (PNN), Support Vector Machine (SVM), MLP, KNN, LR, etc.

Many hybrid methods using filters and wrappers are developed which are proved to be efficient in terms of accuracy, sensitivity and specificity. A two phase feature selection method using filter and wrapper was proposed by (Yuan et al. 1999) In this approach firstly GFSIC (Genetic Feature Selection with Inconsistency Criterion-filter) was used to remove the redundant features and from those they used SBFCV (Sensitivity based Feature selection with V-fold Cross validation- wrapper) to select highly relevant feature. A novel filter-wrapper method was proposed by (Chuang et al. 2008) to find out the minimal gene subset that provides high accuracy. In this model they used Information gain as the filter method and based on the gene scores given by the filter and by fixing the number of features they use PSO base wrapper with SVM as classifier to select the optimal feature subset. They tested this model on microarray classification data. A new hybrid collaborative filter and wrapper method was proposed by (Huang et al. 2007). They developed an approach where two levels of optimization were performed. Outer optimization was performed by the wrapper which selects a subset of features and a filter is applied on the features to perform local optimization. Here Mutual information is used as a filter which performs local optimization. A GRASP (Greedy Randomized Adaptive Search Procedure) algorithm for hybrid feature subset selection in higher dimensional datasets was proposed by (Pablo et al. 2010). GRASP is a multi start constructive method which starts with a random solution and tries to improve in the further stages. A novel feature selection method for diagnosis of erythemato-squamous diseases was proposed by (Xie and Wang, 2010). Here they used F-score filter to preselect the features and those features are passed to SFS (Sequential Feature Selection) based wrapper with SVM as classifier. The efficacy of their proposed model was tested on different combinations of training –testing partitions. A novel feature selection method for biomedical data classification was proposed by (Peng et al. 2010). In the proposed model filters are used for pre-selection of features which are used by wrapper in next phase to select the best optimal feature subset. In this method a new filtering criterion was proposed which considers both discrimination and complimentary capability of the feature. A hybrid model for feature selection was proposed by (Hsu et al. 2010) that use both filters and wrappers. In this approach they used computationally efficient filters to provide the initial subset of features and then wrapper is used to find out the best optimal feature subset from them. The proposed model was tested on microarray cancer data. A new wrapper-filter feature selection method using memetic frame work was proposed by (Zhu et al. 2007). They proposed a wrapper –filter approach in which initial population was initialized with random subset of features and then local improvement process is performed on each subset such that the corresponding feature subset was replaced based on Lamarckian learning. A hybrid wrapper/ filter approach for feature subset selection was proposed by (Prat et al. 2008). In this approach 5 filter approaches (Gini Index, Information Gain, Relevance, Gain ratio, Relief) were used to rank the features in the data set and the highest ranked features are passed to wrapper to find out the optimal feature subset. Here the intention of using filters is to make the life simpler for wrapper by working on less number of features. Feature selection using single / multi objective memetic frameworks were proposed by (Zhu et al. 2009). In this approach they induced filter as local learning procedures of evolutionary algorithms such that they are used to determine whether to add or delete the feature selected by the wrapper. (Hua et al, 2009) analyzed the performance of various feature selection methods in classification of high dimensional data. In this paper they compared the performance of various filter methods like t-test, relief, information gain, etc, when used alone with the models using them with other wrappers. The results obtained depicted that the wrappers when used in two stage process where in first stage filters are used to filter out irrelevant features and wrappers to select best feature subset among the features provided by filters perform well.

4.3 Methodologies used in the proposed model

t-statistic (Fu et al. 2006)

t-statistic is one of the simplest and efficient filter method which ranks the features individually. It is applied only for binary classification datasets. The formula for t-statistic is:

$$t = \frac{|\mu_1 - \mu_2|}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Where µ1, µ2 are the means of given feature values of samples of two classes. n1, n2 are the number of samples of first and second class. σ1 and σ2 represent the standard deviations of the patterns of classes one and two. n1 and n2 represent the number of samples of two classes. 't' value represents the t-statistic value of the feature and the features are ordered in a descending fashion based on 't' value. The features with high't' value are the top ranked features and the features with less t'' value are least ranked features. During feature subset selection the features which are least ranked are eliminated and top ranked features are considered.

Mutual Information Based Feature Selection (MI) (Battiti, 1994)

Mutual Information based feature selection measures the amount of information obtained about one feature by observing another. Let X, Y be two random features, mutual information is defined as

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

H( ) is the entropy of a random feature and measures the uncertainty associated with it. For the continuous feature X, H(X) (similarly for Y also) is defined as

$$H(X) = -\int p(x)\log_2 p(x)dx$$

If X is a discrete random feature, then H(X) is defined as

$$H(X) = -\sum p(X)\log_2 p(X)$$

P(X) is the marginal probability distribution of the random feature X and H(X,Y) is the joint entropy of the random features, is nothing but merely the entropy of their paring H(X,Y), is defined as

$$H(X,Y) = -\sum_{x,y} p(x,y)\log_2 p(x,y)$$

The mutual information approximate algorithm to compute is described as follows:

- *Initialization*: Initialize ‘F’ to set of n features and ‘S’ to an empty set.
- *Computation of the Mutual Information with output class ‘C’*: For each feature f € F we compute I(C;F) as follows:

$$I(C;F) = I(F;C) = \sum_{c,f} P(c,f)\log\frac{P(c,f)}{P(c)P(f)}$$

Where F is feature set, C is output class.

- *Choice of the first feature*: Find the feature f that maximizes I(C; f); set F🡐F\{f}; set S🡐{f}.
- *Greedy selection*: Repeat until |s|=k

  Computation of the MI between features: For all couples of features {f, s} with f € F, s € S, compute I(f; s), if it is already available.

  Selection of the next feature: Choose feature f as the one that maximizes $I(C;f) - \beta\sum_{s\in S} I(f;s)$; Set F🡐F\{f}; Set S 🡐 S U {f}.
- *Output* the set S containing the selected features.

β regulates the relative importance of the Mutual Information between the candidate feature f, the output class C and MI between the candidate feature f, the already selected feature s. When β is zero, I(C; f) alone determines which feature should be selected. If β increases the measure is discounted by the quantity proportional to the total MI with respect to the already selected features (Battiti, 1994).

Multi Layer Perceptron

A multilayer perceptron (MLP) is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate output (wikipedia). It is a supervised learning algorithm with back propagation for training the network. MLP is first trained on historical data so that they learn how to transform input data to a desired response. An MLP consists of single input layer and single output layer and 1 or more hidden layers. MLP's are highly efficient in approximating virtually any input-output map. Multilayer perceptron using a back propagation algorithm is the standard algorithm for any supervised-learning pattern recognition process and the subject of ongoing research in computational neuroscience and parallel distributed processing.

Threshold Accepting Based Logistic Regression (Ravi et al. 2010):

Logistic Regression is used to predict the dependent variable based on continuous or categorical independent variables and also determines the variance of dependent variable explained by independent variables. Logistic Regression has many names in several realms that include logit model, logistic model, and maximum - entropy classifier (Hosmer, David and Stanley, 2000). Logistic Regression applies maximum likelihood estimation after transforming the dependent variable into a logit model to estimate the probability of occurrence of a certain event. Logistic regression is used for only classification problems. Logistic Regression ranks the relative importance of independent variables by associating corresponding weights. Here in our model we used Threshold Accepting Logistic Regression (Ravi et al. 2010). Threshold Accepting algorithm is used to optimize the logistic weights such that minimum mean square error is obtained.

4.4 Hybrid Filter-Wrapper method for Feature Subset Selection

A novel filter-wrapper based hybrid model for feature subset selection had been proposed. Two hybrid wrappers using two classifiers TALR and MLP with binary particle swarm optimization algorithm as the common metaheuristic which wraps around these classifiers had been developed. Figure 13 shows the architecture of proposed model. In the proposed model 10 solutions are fixed for binary PSO in which the first two are initialized using filters. It was made to induce the efficacy of filters into the model and also helps in improving others solutions. Each solution is a feature subset represented in the form of 1's and 0's. If 1 is used, the corresponding feature is considered else if it is 0 then the feature is not considered. The dimensionality of solution is equal to the number of features in the dataset. The process of feature subset selection is as follows:

Initialization

In the proposed model the hybrid wrappers are initialized using binary PSO. The dimension of the solution is equal to the number of features in the dataset. In binary PSO every solution except the first two is initialized to 1 or 0 randomly in all dimensions. To initialize the first two solutions we employed two filters: t-statistic and Mutual Information which provides two sets of prioritized features. The top ranked features (only half the number of features) are considered for initializing the first two solutions of the binary PSO. The rest of the solutions are initialized randomly. For example consider UK Bankruptcy dataset which has 10 features. If the t-statistic ranks the features 1, 3, 6, 7, 8 as top ranked features then the solution1 is initialized as 1010011100. In this way the solutions are initialized for binary PSO and in later iterations they are updated based on the updating formulae of binary PSO. Here classification accuracy calculated on the dataset is considered as the fitness value and the velocity and position values of the solutions are updated such that the fitness value must be maximized.

Feature Subset Selection

Every solution is a feature subset which is carried-over to the classifier. The classifier is trained on the dataset with the features specified in the feature subset. During the training process the classifier tries to learn from the patterns and updates its weights accordingly. After training, the classifier with weights obtained during training process is used for testing. The classifier is made to test on the patterns which are not passed during training process which is referred to as test data and the corresponding classification accuracy is calculated. Classification accuracy specifies that ability of the classifier to learn from the training data. In this way classification accuracy is calculated for all solutions and this classification accuracy is the fitness value for the metaheuristic and is to be optimized i.e., maximized as the iterations progresses. The solution with highest classification accuracy is considered as the global best solution and the highest value obtained by the solution up to that point is considered as the personal best value for that solution. After the algorithm meeting the converging criteria the global best solution is the final feature subset. In our model we proposed two hybrid wrappers which gave two different feature subsets.

Calculating the final feature subset

Two hybrid models have been used in finding the feature subsets which resulted in two feature subsets with different levels of accuracy and in addition to that 10 fold testing is used to calculate the accuracy. We have 10 feature subsets obtained in ten folds from hybrid model-1 and 10 more from hybrid model-2. To find the final feature subset we find the frequency of every individual feature in all ten folds. The frequency count is the number of times the feature appears in the ten feature subsets. The features which have a frequency count of at least 9 (except for Dotcom dataset for which it is 7), i.e., the feature present in all the subsets with a liberty in one subset, are selected. After finding those features we have two sets of features corresponding to two hybrid models. The final feature subset is the set of features which includes all the features present in both the sets obtained from the two models.

4.5 Results and Discussions

The proposed model was tested on 7 well known classification datasets. These of different number of features and patterns and except dotcom and UK bankruptcy dataset all other dataset's are imbalanced datasets. Imbalanced datasets are the datasets where there is no equal proportion of class patterns. All the datasets are binary classification datasets. The details of the datasets are presented in the above section. Tanagra 1.4 was used to employ Mutual Information filter on the datasets which is available at http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html . Table 10 depicts the performance of the hybrid models on different datasets. Hybrid model-1 is the hybrid wrapper model with TALR as the classifier and hybrid model-2 is the hybrid wrapper model with MLP as classifier. The classification accuracy obtained is the average value of the ten folds. The number of features selected by the models was fixed to the half the total number of features in the dataset.

One of the biggest challenges faced by many researchers in solving feature selection problem is about finding the final feature subset to consider. Table 11 depicts the features selected by the hybrid models 1 and 2. Here 'A' column contains the features which have specified frequency count (for all it is at least 9 except for dotcom dataset it is 7) from hybrid model 1 similarly 'B' corresponds to hybrid model 2. The final feature subset is the set that includes the features of both 'A' and 'B'. Consider Turkish dataset which contain 12 features in original dataset. According to hybrid model-1 F5, F6, F11 are the features which are more common in all features subsets of ten folds. Similarly F2, F8, F10, F11 are the features that occur in most of the feature subsets of ten folds for hybrid model-2. The final feature subset is F2, F5, F6, F8, F10, and F11 which is a union set of both the feature sets obtained from two hybrid models. In this way final feature subset is calculated for all datasets.

4.6 Conclusions

A novel approach for feature subset selection for binary classification problems which uses the potentiality of wrapper and filters had been proposed. During our experiments we observed that not only the dataset used for training, the classifier that we use in our model also highly influences the result obtained using the model. We are also successful in providing efficient results with fewer features (half of the total features for all datasets). The efficacy of our proposed model is tested by testing its performance on large data sets with more features where our model produced better results by considering half of the features and also all the accuracies are the average of ten folds. The proposed model gives a final feature subset that considers the efficacies of both hybrid models.

5

Overall Conclusions

Designing hybrid methodologies that provide optimal solution is a challenging and enthusiastic task. As the dimensionality of the problem increases the complexity of the problem increases which hurdles finding the optimal solution. So any hybrid methodology developed must provide an optimal solution considering all the factors: dimensionality, functional evaluations, accuracy, etc.

We developed several hybrid methodologies and presented their efficacy on different benchmark problems and datasets. These methodologies not only provide optimal solution but also perform better than the traditional methods. These hybrids provide more feasibility to the users in choosing the methods for the problems.

In the first part a novel metaheuristic using PSO embedded with TA was developed. Its efficacy was proven when we tested it on 34 benchmark problems. When compared with traditional PSO algorithm the proposed hybrid outperformed in terms of accuracy, functional evaluations, success rate, standard deviations. This outcome is the result of employing TA to tune the worst solution probabilistically.

In the second part two other novel Metaheuristics using harmony search were developed. At first we developed a MHS+MGDA hybrid where we first employed modified harmony search to find the near optimal solution and then we pass the best solution to MGDA which fine tune the solution to provide the final optimal solution. We also developed another hybrid methodology using MHA and TA. The developed methodology outperformed MHS and MHS+MGDA in terms of accuracy and functional evaluations on 20 benchmark problems. The developed hybrids provided better results because the harmony search provides efficient improved solution at the beginning of the search but as the search prolongs it becomes rigid and retards in providing optimal solution.

In the final part of my project a novel filter-wrapper based method for feature subset selection method was developed. Two hybrid wrappers with binary PSO as the background optimization algorithm was developed and a final feature subset with features that are more frequent in the hybrid wrappers are considered. The developed model was tested on 7 most popular benchmark datasets are developed. The proposed hybrid produced high classification accuracies with only half the features and the accuracies are all the average of ten folds.

Bibliography

Ali MM, Charoenchai K, and Zelda BZ (2005). A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems. Journal of Global optimization, Vol.31, pp. 635-672.

Arun N, Ravi V (2009). An ant colony optimisation and nelder-mead simplex search hybrid algorithm for unconstrained optimization. In: Proceedings of the International Workshop on Machine Intelligence Research (MIR Day, GHRC-Nagpur) pp.25–31.

Axis Parallel Hyper-Ellipsoidal Problem: Global optimum www document [cited 2010 Nov 19] available from: http://www.geatbx.com/docu/fcnindex-01.html.

Basiri ME, Aghaee NG and Aghdam MH (2008). "Using Ant Colony Optimization-Based Selected Features for Predicting Post-synaptic Activity in Proteins ," Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, Lecture Notes in Computer Science, Vol. 4973/2008, pp.12-23.

Battiti R (1994). "Using mutual information for selecting features in supervised neural net learning," IEEE Trans. Neural Netw., vol. 5, Issue 4, pp. 537–550.

Ben N and Li L (2009). A Novel PSO –DE based Hybrid Algorithm for Global Optimization. Advanced intelligent computing Theories and Applications with Aspects of Artificial Intelligence; vol.5227 pp. 156-163.

Bermejo P, Gamez JA, Pureta JM (2011). "A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets," Pattern Recognition Letters, Vol.32, Issue 5.

Bhat TR, Venkataramani D, Ravi V and Murty CVS (2006). Improved differential evolution method for efficient parameter estimation in biofilter modeling, Biochemical Engineering Journal, Vol. 28, pp.167-176.

Beynon MJ, and Peel MJ (2001). Variable Precision Rough Set Theory and Data Discretisation: An Application to Corporate Failure Prediction, OMEGA - International Journal of Management Science, Vol.29, Issue 6, pp.561-576.

Bohachevsky ME., Johnson ME., and Stein ML (1986). Generalized simulated annealing for function optimization. Techno metrics, Vol.28, pp.209-217.

Bose I, (2006). Deciding the financial health of dot-coms using rough sets, Information and Management, Vol.43, pp.835-846.

Bose I & Pal R (2006). Predicting the survival of or failure of click-and-mortar corporations: a knowledge discovery approach, European Journal of Operational Research, Vol.174, pp. 959-982.

Canbas SC, Kilic SB (2005). Prediction of commercial bank failure via multivariate statistical analysis of financial structures: The Turkish case," European Journal of Operational Research, Vol.166, Issue 2, pp.528-546.

Chandra DK, Ravi V, Bose I (2009). Failure prediction of dotcom companies using hybrid intelligent techniques, Expert Systems with Applications, Vol. 36, Issue 3, Part 1, pp. 4830-4837.

Chauhan N and Ravi V (2010). Differential Evolution and Threshold Accepting Hybrid Algorithm for Unconstrained Optimization. International Journal of Bio-Inspired Computation, Vol.2, pp.169 – 182.

Chelouah R and Siarry P (2003). Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multi-minima functions, European Journal of Operational Research, Vol.148, pp.335-348.

Chelouah R, Siarry P (2005) A hybrid method combining continuous tabu search and nelder-mead simplex algorithms for the global optimization of multiminima functions. European Journal of Operational Research, Vol.161, pp.636–654.

Chen LH, Hsiao HD (2008). Feature selection to diagnose a business crisis by using a real GA-based support vector machine: an empirical study. Exp Syst Appl, Vol.35, pp.1145–1155.

Chuang IY, Ke CH, Yang CH (2008). A hybrid both filter and wrapper feature selection method for microarray classification, Proceedings of the International MultiConference of Engineers and Computer Scientists 2008, Hong Kong, Vol.1, pp.19-21.

David HW & Stanley L (2000). "Applied logistic regression," (2nd edition), Newyork, Chichester, Wiley.

Deep K and Bansal JC (2009). Hybridization of Particle Swarm Optimization with Quadratic Approximation. Opsearch, Springer Vol.46, Issue 1, pp.3-24.

Dekkers L, and Szego G (1991). Global Optimization and Simulated Annealing. Mathematical Programming, Vol.50, pp.367-393.

Dixon L and Szego G (1978). Towards Global Optimization Vol. 2, North Holland, New York.

Dueck G and Scheur T (1990). Threshold Accepting: A General Purpose Optimization Algorithm appearing Superior to Simulated Annealing. Journal of Computational Physics, Vol.90, pp.161-175.

Dueck G (1993). New optimization heuristics. Journal of Computational Physics Vol.104, Issue 1, pp.86–92.

Dorigo M (1992). Optimization, Learning and Natural algorithms, PhD Thesis, Politecnico di Milano, Italy.

Dorigo M, Maniezzo V and Colorni A (1996). Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics – Part B, Vol.26, Issue 1, pp.29–41.

Dorigo M and Stützle T (2004). Ant Colony Optimization. MIT Press, 2004, Cambridge, MA.

Edmund KB and Graham K (2005). Search Methodologies: Introductory Tutorials in Optimization and Decission Support Techniques, Springer.

Fan SKS and Zahara E (2007). A Hybrid Simplex Search and Particle Swarm Optimization for Unconstrained Optimization European Journal of Operational Research, Vol.181, pp. 527–548.

Fesanghary M, Mahdavi M, Joldan MM, Alizadeh Y (2008). Hybridizing harmony search algorithm with sequential programming for engineering optimization problems, Comp. Methods Appl. Mech. Eng., Vol.197, pp.3080-3091.

Frank A & Asuncion A (2010). UCI Machine Learning Repository[http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Fu X, Tan F, Wang H, Zhang YQ, & Harrison R (2006). Feature similarity based redundancy reduction for gene selection, In proceedings of 2006 international conference on data mining (DMIN'06) Las Vegas, pp. 26-29.

Huang J, Cai Y, Xu X (2007). A hybrid genetic algorithm for feature selection wrapper based on mutual information, Pattern Recognition Letters, Vol. 28, Issue 13, pp. 1825-1844.

Hsu HH, Hsieh CW, Lu MD (2011). "Hybrid feature selection by combining filters and wrappers," Expert Systems with Applications, Vol. 38, Issue 7, pp. 8144-8150.

Holland JH (1962). Outline for logical theory of adaptive systems. Journal of ACM, Vol.3, pp. 297-314.

Gao XZ, Wang X, Ovaska J (2009). Uni-Modal and Multi Modal optimization using modified harmony search methods, IJICIC, Vol.5, Issue 10(A), pp.2985-2996.

Glover F (1989). Tabu Search, part I. ORSA Journal on computing, Vol.1 Issue 1 pp.190-206.

Glover F (1990). Tabu search, Part II. ORSA Journal on Computing, Vol.2, Issue 1, pp.4–32.

Geem Z, Kim J, Loganathan G (2001). A new heuristic optimization algorithm: harmony search. Simulation, Vol.76, pp.60–68.

Holland JH (1975). Adaptation in natural and artificial systems. University of Michigan press, Ann Harbor, MI, Internal Report.

Jansson C and Knu¨ ppel O (1995). A Branch and Bound Algorithm for Bound Constrained Optimization Problems without Derivatives. Journal of Global Optimization, Vol.7, pp. 297–331.

Jiang Y, Tiesong Hu, Huang CC and Wu X (2007). An Improved Particle Swarm Optimization Algorithm. Applied Mathematics and Computation, Vol.193, pp. 231-239.

Jua J, Tembe WD, Dougherty ER (2009). Performance of feature-selection methods in the classification of high-dimension data," Pattern Recognition Vol.42, pp.409-424.

Kabir MM, Islam MM, Murase K (2010). A new wrapper feature selection approach using neural network," Neurocomputing, Vol.73, Issues 16-18, pp. 3273-3283, 10th Brazilian Symposium on Neural Networks (SBRN2008).

Kaveh A, Talatahari S (2009). PSO, ant colony strategy and harmony search scheme hybridized for optimization of truss structures, Computers and Structures, Vol.87, pp.267-283.

Kennedy J and Eberhart R (1995). Particle Swarm Optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp.1942–1948.

Kirkpatrick S, Gelatt CDJ, Vecchi, MP (1983). Optimization by simulated annealing. Science, Vol.4598, pp.671 – 680.

Kohavi R and John GH (1997). Wrappers for feature subset selection, Artificial Intelligence, Vol. 97, Issues 1-2, pp. 273-324.

Kulakarni R. and Venayagamoorthy GK (2007). An Estimation of Distribution Improved Particle Swarm Optimization Algorithm. Intelligent Sensors, Sensor Networks and Information Processing Conference, Melbourne.

Levy AV and Montalov A (1985). The Tunneling Algorithm for the Global Minimization of Functions. Society for Industrial and Applied Mathematics, Vol.6, pp: 15-29.

Li H, Li L (2007). A novel hybrid particle swarm optimization algorithm combined with harmony search for higher dimensional optimization problems, International conference on Intelligent Pervasive Computing, Jeju Island, Korea.

Li WT, Shi L, Xu XW, and Hei YQ (2008). Improved GA and PSO Culled Hybrid Algorithm for Antenna Array Pattern Synthesis. Progress in Electromagnetic Research, PIER, Vol.80, pp.461-476.

Lin SW, Ying KC, Chen SC and Lee ZJ (2008). Particle swarm optimization for parameter determination and feature selection of support vector machines, Expert Systems with Applications Vol.35, Issue 4, pp.1817-1824.

Michalewicz Z (1996). Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, Berlin/Heidelberg/New York.

More´J., Garbow B. and Hillstrom K (1981). Testing Unconstrained Optimization Software. ACM Transaction on Mathematical Software, Vol.7, pp.17–4.

Neumaier A Global and local optimization [cited 2010 Nov 20]. www-document available at http://solon.cma.univie.ac.at/~neum/glopt.html.

Noelia S, Amparo AB, and Maria TS (2007). Filter methods for feature selection – a comparative study," IDEAL 2007, LNCS 4881, pp. 178–187.

Olmeda I, Fernandez E (1997). Hybrid classifiers for financial multicriteria decision making: the case of bankruptcy prediction, Computational Economics, Vol.10, no.4, 1997, pp. 317-335.

Pan QK, Suganthan PN, Liang JJ, Tasgetiren MF (2011). A local best harmony search algorithm with dynamic sub-harmony memories for lot-streaming flow shop scheduling problem, Expert systems with applications, Vol.38, pp.3252-3259.

Pant M and Radha T (2009). Particle Swarm Optimization with Crossover Operator and its Engineering Applications. IAENG International Journal of Comp. Science, Vol.36 Issue 2.

Price WL (1977). Global Optimization by Controlled Random Search. Computer Journal, Vol.20, pp.367–370.

Pendharkar PC, Rodger JA (2004). An empirical study of impact of crossover operators on the performance of non-binary genetic algorithm based neural approaches for classification," Computer Operational Research Vol.31, pp. 481–498.

Peng Y, Wu Z, Jiang J (2010). A novel feature selection for biomedical data classification," Journal of Biomedical Informatics, Vol. 43, Issue 1, pp. 15-23.

Prati RC, Batista GEAPA, Monard MC (2008). A hybrid wrapper / filter approach for feature subset selection," Electronic Journal of SADIO, Vol.8, Issue 1, pp. 12-24.

Ravi V, Murthy BSN, and Reddy, PJ (1997). Non-equilibrium simulated annealing-algorithm applied to reliability optimization of complex systems. IEEE Transactions on Reliability, Vol.46, pp.233-239.

Ravi V (2004). Optimization of Complex System Reliability by a Modified Great Deluge Algorithm. Asia-Pacific Journal of Operational Research Vol.21 Issue 4 pp. 487–497.

Ravi v, Shalom SAA, Manickavel A (2004). Sputter process variable prediction via data mining," Cybernetics and Intelligent Systems, 2004 IEEE Conference, Vol.1, pp.256 – 261.

Ravi v and Pramodh C (2010). Non-linear principal component analysis-based hybrid classifiers: an application to bankruptcy prediction in banks, International Journal of Information and Decision Sciences, Vol. 2, Issue 1, pp. 50 – 67.

Ravisankar P, Ravi V, Bose I (2010). Failure prediction of dotcom companies using neural network-genetic programming hybrids, Information Sciences, Vol.180, Issue 8, pp.1257-1267.

Salhi S, Queen, NM (2004). A hybrid algorithm for identifying global and local minima when optimizing functions with many minima, European Journal of Operational Research, Vol.155, pp.51–67.

Salomon R (1995). Reevaluating Genetic Algorithms Performance under Co-ordinate Rotation of Benchmark Functions. Bio Systems Vol.39 Issue 3 pp. 263-278.

Schwefel HP (1995). Evolution and Optimum Seeking, John Wiley and Sons, New York.

Schimdt H and Thierauf G (2005). A Combined Heuristic Optimization Technique. Advance in Engineering Software, Vol.36, Issue 1 pp.11-19.

Shelokar PS, Siarry P, Jayaraman VK and Kulkarni BD (2006). Particle Swarm and Ant Colony Algorithms Hybridized for Improved Continuous Optimization, Applied Mathematics and Computation Journal, Vol.188, pp.129–142.

Sphere problem; global and local optima [cited on 2010 Nov 20] available at: www.optima.amp.i.kyoto-.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page113.htm

Srinivas M. and Rangaiah (2007). Differential Evolution with Tabu list for Global Optimization and its Application to Phase Equilibrium and Parameter Estimation. Problems Ind Engg Chem. Res., Vol.46, pp.3410-3421.

Storn R. and Price K (1997). Differential Evolution: A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization, Vol.11, pp: 341–359.

Tasgetiren MF, Liang YC (2004). A binary particle swarm optimization algorithm for lotsizing problem," Journal of Economic and Social Research, Vol.5, Issue 2, pp.1-20.

Thomas LC, Edelman DB, and Crook JN (2002). Credit scoring and its applications, Philadelphia, Siam.

Torn A and Zilinskas A (1989). Global Optimization. Lecture Notes in Computer Science, Vol. 350, Springer-Verlag, Berlin/Heidelberg/New York.

Trafalis TB and Kasap S (2002). A novel metaheuristics approach for continuous global optimization, Journal of global optimization, Vol.23, pp.171-190.

Verikas A, Kalstye Z, Bacauskiene M, Gelzinis A (2010). Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: a survey," Soft Computing, Vol.14, Issue 19, pp.995-1010.

Wolfe MA (1978). Numerical Methods for Unconstrained Optimization, Van Nostrand Reinhold Company, New York.

Xie J, Wang C (2011). Using support vector machines with a novel hybrid feature selection method for diagnosis of erythemato-squamous diseases, Expert Systems with Applications, Vol. 38, Issue 5, pp.5809-5815.

Yuan H, Tseng SS, Gangshan W, Fuyan Z (1999) A two-phase feature selection method using both filter and wrapper" Systems, Man, and Cybernetics, IEEE SMC'99 conference proceedings, Tokyo, Japan, Vol.2, pp. 132-136.

Zakharov Problem Global and local optima [cited on 2010 Nov 20] available at:
www.optima.amp.i.kyoto-ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page3088.htm

Zhang W and Xie XF (2003). DEPSO: Hybrid Particle Swarm with Differential Evolution Operator. IEEE International Conference on Systems, Man & Cybeetics (SMCC), Washington DC, USA.

Zhao Z and Liu H (2007). Searching for interacting features, In proceedings of 20th International Joint Conferences on Artificial Intelligence, Hyderabad, India.

Zhu Z, Ong YS, Dash M (2007). Wrapper-filter feature selection algorithm using memetic framework, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol.37, Issue 1, pp.70-76.

Zhu Z, Ong YS, and Kuo JL (2009). Feature selection using single/multi-objective memetic frameworks," Studies in Computational Intelligence, Vol. 171/2009, pp.111-131.

A

Description of datasets used for Feature subset Selection

Spanish Banks Dataset

The “Spanish banks” data is obtained from (Olmeda and Fernandez 1997). ”Spanish banks” dataset contains the list of banks which were bankrupt and non-bankrupt, so the target variable class contains two classes: bankrupt and non-bankrupt. Spanish banking industry suffered the worst crisis during 1977-85 resulting in a total cost of 12 billion dollars. The ratios used for the failed banks were taken from the last financial statements before the bankruptcy was declared and the data of non-failed banks was taken from 1982 statements. This dataset contains 66 banks where 37 went bankrupt and 29 healthy banks. It contains 9 financial ratios and a target class variable. Table 12 describes the financial ratios of Spanish bank dataset.

Turkish Banks Dataset

“Turkish banks” dataset is a bankruptcy prediction dataset where it contains patterns of several banks in which some are bankrupt and some other are non-bankrupt. It was obtained from (Canbas & Kilic 2005), which is available at (http://www.tbb.org.tr/english/bulten/yillik/2000/ratios.xls). Canbas & Kilic (2005) chose only 12 ratios as the early warning indicators that have the discriminating ability (i.e. significant level is <5%) for healthy and failed banks one year in advance. Among these variables, 12th variable has some missing values meaning that the data for some of the banks are not given. So, we filled those missing values with the mean value of the variable, is a general approach in data mining. This dataset contains 40 banks where 22 banks went bankrupt and 18 banks are healthy. Table 13 describes the financial ratios of Turkish bank dataset.

UK Bankruptcy Dataset

The "UK bankruptcy" dataset is taken from Beynon and Peel (2001). This dataset contains 60 patterns among which 30 are healthy and 30 bankrupt. Each pattern corresponds to each bank. The dataset contains 10 financial ratios and 1 target class variable. Table 15 contains the financial ratios of UK Bankruptcy dataset.

Dotcom companies Dataset

"Dotcom companies" dataset consists of 240 patterns of click- and-mortar companies and for each pattern there are 24 financial ratios. It was take from Wharton Research Data service for the year 2000. Dotcom data set was first analyzed by Bose (2006) followed by Bose and pal (2006), Chandra et al. (2010), Ravisankar et al.(2010). Out of 240 patterns 120 patterns are about firms that failed and the rest 120 did not fail. Out of 24 financial ratios first fifteen were the most popularly used ones in literature related to bankruptcy prediction of financial firms and the rest were constructed to capture the novelty of dotcom companies, and the rest of the features were used to reflect their sales, earnings, cash, income, market capitalization and stock prices. Financial Ratios used in Dotcom data set are presented in table 16.

UK Credit data set

UK credit dataset consists of 1225 patterns of the customers applied for credit product. It contains 14 financial ratios regarding the applicants and in those we removed the 3 features: phon, aes, res which corresponds to "presence of landline or not", "applicant's employment status", "residential status" of the applicant. They are removed as some are very irrelevant and other has many categorical values. In 12225 patterns 323 are of bad customers i.e., customers with very less credit and 902 are of good customers. The financial ratios of UK Credit dataset are tabulated in table 14.

Pima Indians Dataset

Pima Indians dataset was taken from National Institute of Diabetes and Digestive and Kidney Diseases in the year 1990. It was available at "http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes". It contains of 768 patterns of the patients in which 268 are tested positive for diabetes and 500 of the patients who are tested negative. It contains 8 features and one target class variable. Table 17 describes the features of pima Indians dataset.

Spectf Dataset

Spectf dataset contains data on cardiac Single Proton Emission Computed Tomography (SPECT) images. Here each patient classified into two categories: normal and abnormal. .Kurgan and Cios are the donors of this dataset. It contains 267 SPECT image sets (patients) which were processed to extract features that summarize the original SPECT images. As a result, 44 continuous feature patterns were created for each patient. Table 18 contains the features of *Spectf* dataset.

B

Unconstrained optimization benchmark problems used for testing the proposed models

1. Sphere Problem:

Sphere function is defined as

$$\min_x f(x)=\sum_{i=1}^{n} x_i^2$$

where n is the number of variables and its search domain ranges from -5.12 ≤ xi ≤ 5.12, i = 1,2,..n. There are no local minima and has a global minima x*= (0,0,….0) , f(x*)=0.

2. *Rosenbrock's function* (Schwefel, 1995).

Minimize $$f(x)=\sum_{i=1}^{n} 100(x_{i+1}-x_i^2)^2+(1-x_i)^2; -2.048 \le x_i \le 2.048$$

It is a classic optimization problem, also known as Banana function. The global optimum is inside a long, narrow, parabolic shaped flat valley. To find the valley is trivial, however convergence to the global optimum is diffcult. The global minimum occurs at (1, 1, …,1) with objective function value of 0. We tested the two dimensional version.

3. *Goldstein-Price* (Dixon and Szego, 1978)

Minimize

$$f(x)=[1+(x_1+x_2+1)^2.(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]$$
$$\times[30+(2x_1-3x_2)^2.(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]; -2 \le x_1, x_2 \le 2$$

There are four local minima and the global minima occurs at (0,-1) with f(0,-1) = 3.

4. Schaffer1 problem (SF1) (Michalewicz, 1996):

$$\min_x f(x)=0.5+\frac{(\sin\sqrt{(x_1^2+x_2^2)})^2-0.5}{1+0.001(x_1^2+x_2^2)^2}$$

subject to -100 ≤ x_1,x_2 ≤ 100 , the number of local minima is not known but global minima is located at x* = (0,0) with f(x*) = 0.

5. Schaffer2 problem (SF2) (Michalewicz, 1996):

$$\min_x f(x) = (x_1^2 + x_2^2)^{0.25} (\sin^2(50((x_1^2 + x_2^2)^{0.1}) + 1)$$

subject to $-100 \leq x_1, x_2 \leq 100$, the number of local minima is not known but global minima is located at x* = (0,0) with f(x*) = 0.

6. *Bohachevsky 1 Problem (BF1)* (Bohachevsky et al., 1986)

$$\min_x f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.7$$

Subject to $-50 \leq x_1, x_2 \leq 50$. The number of local minima is unknown but the global minimizer is located at $x^* = (0,0)$ with $f(x^*) = 0$.

7. *Bohachevsky 2 Problem (BF1)* (Bohachevsky et al., 1986)

$$\min_x f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3$$

Subject to $-50 \leq x_1, x_2 \leq 50$. The number of local minima is unknown but the global minimizer is located at $x^* = (0,0)$ with $f(x^*) = 0$.

8. Periodic Problem (PRD) (Price, 1977):

$$\min_x f(x) = 1 + \sin^2 x_1 + \sin^2 x_2 - 0.1\exp(-x_1^2 - x_2^2)$$

subject to $-10 \leq x_1, x_2 \leq 10$. There are 49 local minima all with minimum values 1 and global minimum located at x* = (0, 0) with f(x*)=0.9.

***9. Camel Back – 6 Six Hump Problem (CB3)* (Dixon and Szego, 1978; Michalewicz, 1996)**

$$\min_x f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$$

Subject to $-5 \leq x_1, x_2 \leq 5$. This function is symmetric about the origin and has three conjugate pairs of local minima with values $f \approx$ -1.0316, -0.2154, 2.1042. The function

$\approx$ -1.0316.

10. *Becker and Lago Problem (BL) (Price, 1977)*

$$\min_x f(x) = (|x_1| - 5)^2 + (|x_2| - 5)^2$$

Subject to $-10 \leq x_1, x_2 \leq 10$. The function has four minima located at ($\pm 5, \pm 5$), all with $f(x^*) = 0$.

***11. Ackley's Problem (ACK)* (Storn and Price, 1997)**

$$\min_x f(x) = -2\exp\left(-0.02\sqrt{n^{-1}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(-0.02\sqrt{n^{-1}\sum_{i=1}^{n}\cos(2\pi x_i)}\right) + 20 + e$$

Subject to $-30 \leq x_i \leq 30$, $i \in \{1,2,\ldots,n\}$. The number of local minima is not known. The global minimum is located at the origin with $f(x^*) = 0$. Tests were performed for n=10.

12. Salomon Problem (SAL) (Salomon,1995)

$$\min_x f(x) = 1-\cos(2\pi \| x \|) + 0.1 \| x \|$$

where $\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2}$ and $-100 \le x_i \le 100$. The number of local minima is not known but global optimum lies at x*=(0,0,...0) with f(x*)=0 and *n* is the number of variables.

13. Kowalik Problem (KL) (Jansson and Knuppel, 1995)

$$\min_x f(x) = \sum_{i=1}^{11}\left(a_i - \frac{x_1(1+x_2 b_i}{1+x_3 b_i + x_4 b_i^2}\right)^2$$

subject to $0 \le x_i \le 0.42$. The global optimum lies at x* ≈ (0.192,0.190,0.123,0.135) and $f(x^*) \approx 3.0748 * 10^{-4}$.

The values for a_i, b_i constants are:

a[1-11]={0.1957,0.1947,0.1735,0.16,0.0844,0.0627,0.0456,0.0323,0.0235,0.0246}

b[1-11]={0.25,0.5,1.0,2.0,4.0,6.0,8.0,10.0,12.0,14.0,16.0}

14. Levy and Montalvo 1 Problem: (LM1) (Levy and Montalvo, 1985)

$\min_x f(x) =$

$$(\pi / n)\left(10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})] + (y_n-1)^2\right) where yi = 1+1/4(x_i+1)$$

Where $y_i=1/4(x_i+1)$ for $-10 \le x_i \le 10$. i € {1,2,...n}. There are 5^n local minima and global minima is f(x*) = 0 with x* = (-1,-1, ...-1). Here '*n*' is the number of variables.

15. Levy and Montalvo 2 Problem: (LM1) (Levy and Montalvo, 1985; Dekker and Aarts, 1991)

$$\min_x f(x) = 0.1\left(\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})] + (x_n-1)^2[1+\sin^2(2\pi x_n)]\right)$$

subject to $-5 \le x_i \le 5$, i €{1,2, ..*n*} . There are approximately 15n local minima and global optimum is f(x*)=0 at x*= (1,1, .. 1). Here '*n*' is the number of variables.

16. Meyer and Roth Problem (MR) (Wolfe, 1978)

$$\min_x f(x) = \sum_{i=1}^{5}\left(\frac{x_1 x_3 t_i}{(1+x_1 t_1 + x_2 v_i)} - y_i\right)^2$$

subject to $-10 \le x_i \le 10$, i €{1,2, ..*n*} . This is a least squares problem with minimum value $f(x^*)=0.4 * 10^{-4}$ located at x*= (3.13, 15.16, 0.78).

data:

t[1-5]={1.0,2.0,1.0,2.0,0.1}

v[1-5]={1.0,1.0,2.0,2.0,0.0}

y[1-5]={0.126,0.219,0.076,0.126,0.186}

17. Miele and Cantrell Problem (MCP) (Wolfe,1978):

$$\min_x f(x)= (\exp(x_1)-x_2)^4+100(x_2-x_3)^6+(\tan(x_3-x_4))^4+x_1^8$$

subject to $-1 \leq x_i \leq 1$, i € {1,2,3,4} . The number of local minima is unknown but the global optimizer is located at x*={0,1,1,1} with *f*(x*)=0.

18. *Neumaier2 Problem (NF2)* (Neumaier, 2003b)

$$\min_x f(x)=\sum_{k=1}^{n}(b_k-\sum_{i=1}^{n}x_i^k)^2$$

Subject to $0 \leq x_i \leq n$, $i \in \{1,2,\ldots,n\}$. The case considered here is n=4 and b={8,18,44,114}. The global minimum is *f*(1,2,2,3)=0.

19. Powell's Quadratic Problem (PWQ) (Wolfe, 1978)

$$\min_x f(x)= (x1+10x1)2+5(x_3-x_4)^2+(x_2-2x_3)^4+10(x_1-x_4)^4$$

subject to $-10 \leq x_i \leq 10$,i € {1,2,3,4} . This is a unimodal function with *f*(x*)=0, x*=(0,0,0,0). This minimiser is difficult to obtain with accuracy as the hessian matrix at the optimum is singular.

20. Wood's function (WF) (Michalewicz, 1996; Wolfe 1978)

$$\min_x f(x)= 100(x2-x_1^2)^2+(1-x_1)^2+90(x_4-x_3^2)^2+(1-x_3)^2+10.1\left[(x_2-1)^2+(x_4-1)^2\right]+19.8(x_2-1)(x_4-1)$$

subject to $-10 \leq x_i \leq 10$, i € {1,2,3,4} . The function has a saddle near (1, 1, 1, 1). The only minimum is located at x* = (1, 1, 1, 1) with *f*(x*)=0.

21. *Zakharov (Z_n) (n variables)* (Chelouah and Siarry, 2003)

$$Z_n(x) = (\sum_{j=1}^{n}x_j^2)+(\sum_{j=1}^{n}0.5jx_j)^2+(\sum_{j=1}^{n}0.5jx_j)^4;$$

5 functions were considered: Z_2, Z_5, Z_{10}, Z_{50} and Z_{100}; search domain: $-5 \leq x_j \leq 10$, j = 1,…,n;n several local minima (exact number unspecified in usual literature);

1 global minimum: x* = (0,…,0); Z_n (x*) = 0.

22. Axis Parallel Hyper-ellipsoidal Function

$$\min_x f(x)= \sum_{i=1}^{n}ix_i^2$$

subject to $-5.12 \leq x_i \leq 5.12$. It is also known as weighted sphere model. It is continuous, convex and unimodal. It has a global minimum at *f*(x*)=0 at x*={0,0,…0}.

23. Rastrigin Problem (RG) (Storn and Price, 1997; Torn and Zilinskas, 1989)

$$\min_x f(x)= 10n+\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)\right]$$

subject to -5.12 $\leq x_i \leq$ 5.12 , i € {1,2,...n}. The total number of minima for this function is not exactly known but global optimizer is located at x* = {0,0,...0} with f(x*)=0.

24. ***Shekel 5 Problem (S5)*** **(Dixon and Szegö, 1978)**

$$\min \; f(x) = -\sum_{i=1}^{5} \frac{1}{\sum_{j=1}^{4}(x_j - a_{ij})^2 + c_i}$$

Subject to 0 ≤ x_j ≤10, j ∈ {1, 2, 3, 4} with constants a_{ij} and c_j given in the following table. There are five local minima and the global minimiser is located at x* = (4.00, 4.00, 4.00, 4.00) with f(x*) ≈ -10.1499.

Table Data for Shekel 5, Shekel 7, Shekel 10 problem.

	i	aij				ci
		j=1	j=2	j=3	j=4	
S5	1	4	4	4	4	0.1
	2	1	1	1	1	0.2
	3	8	8	8	8	0.2
	4	6	6	6	6	0.4
	5	3	7	3	7	0.4
S7	6	2	9	2	9	0.6
	7	5	5	3	3	0.3
S10	8	8	1	8	1	0.7
	9	6	2	6	2	0.5
	10	7	3.6	7	3.6	0.5

25. ***Shekel 7 Problem (S7)*** **(Dixon and Szegö, 1978)**

$$\min \; f(x) = -\sum_{j=1}^{7} \frac{1}{\sum_{i=1}^{4}(x_j - a_{ij})^2 + c_i}$$

Subject to 0 ≤ x_j ≤ 10, j ∈ {1,2,3,4} with constants a_{ij} and c_j given in above table. There are seven local minima and the global minimiser is located at x* = (4.00, 4.00, 4.00, 4.00) with f(x*) ≈ -10.3999.

26. ***Shekel 10 Problem (S10)*** **(Dixon and Szegö, 1978)**

$$\min \; f(x) = -\sum_{j=1}^{10} \frac{1}{\sum_{i=1}^{4}(x_j - a_{ij})^2 + c_i}$$

Subject to 0 ≤ x_j ≤ 10, j ∈ {1,2,3,4} with constants a_{ij} and c_j given in table. There are 10 local minima and the global minimiser is located at x* = (4.00, 4.00, 4.00, 4.00) with f(x*) ≈ -10.5319.

27. Aluffi - Pentini's problem (Aluffi-Pentini et al., 1985)

$$\min f(x) = 0.25x_1^4 - 0.5x_1^2 + 0.1x_1 + 0.5x_2^2$$

Subject to -10 ≤x_1,x_2≤ 10. The function has two local minima, one of them is global with f(x*) ≈ -0.3523 located at (-1.0465, 0).

28. Dekker and Aarts Problem (DA) (Dekkers and Aarts, 1991)

$$\min f(x) = 10^5 x_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + 10^{-5} + (x_1^2 + x_2^2)^4$$

Subject to -20≤ x_1, x_2≤ 20. The origin is a local minimizer, but there are two global minimizers located at x* =(0,15) and (0,-15) with f(x*) = -24777.

29. Easom Problem (EP) (Michalewicz, 1996)

$$\min f(x) = -\cos(x_1)\cos(x_2)\exp\left(-(x_1-\pi)^2-(x_2-\pi)^2\right)$$

Subject to -10≤ x_1, x_2≤10. The minimum value is located at (π, π) with f(x*) = -1. The function value rapidly approaches zero, when away from (π, π).

30. Hartman 3 Problem (H3) (Dixon and Szego, 1978)

$$\min f(x) = -\sum_{i=1}^{4} c_i exp\left[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right]$$

Subject to 0 ≤ x_j ≤ 1, j € {1, 2, 3} with constants a_{ij}, p_{ij} and c_i given in the following table. There are four local minim, x_{local} ≈ (p_{i1}, p_{i2}, p_{i3}) with f(x_{local}) ≈ -ci. The global minimum is located at x* ≈ (0.114614, 0.555649, 0.852547) with f(x*) ≈ -3.862782.

Table: Data for Hartman 3 problem

i	ci	aij			pij		
		j=1	j=2	j=3	j=1	j=2	j=3
1	1	3	10	30	0.3689	0.117	0.2673
2	1.2	0.1	10	35	0.4699	0.4387	0.747
3	3	3	10	30	0.1091	0.8732	0.5547
4	3.2	0.1	10	35	0.03815	0.5743	0.8828

31. Modified RosenBrock Problem (MRP) (Price, 1977)

$$\min f(X) = 100(x_2 - x_1^2)^2 + [6.4(x_2-0.5)^2 - x_1 - 0.6]^2$$

Subject to -5≤ x_1, x_2≤5. This function has two global minima each with f(x*) = 0 (Corresponding to the intersection of two parabolas) and a local minimum. The global minima are located at x* ≈ (0.3412, 0.1164), (1, 1).

32. Paviani Problem (PP) (Himmelblau, 1972)

$$\min f(\mathrm{X}) = \sum_{i=1}^{10}[(\ln(x_{i-2}))^2 + (\ln(10-x_i))^2] - (\prod_{i=1}^{10} x_i)^{0.2}$$

Subject to 2 ≤ x_i ≤10, i € {1, 2, 3}. This function has a global minimize at x_i^* ≈ 9.351 for all I, with f(x*) ≈ -45.778.

Tables

Table 1: Parameters fixed for the PSOTA hybrid

SNo	Parameter name	Value
For PSO part		
1	C1	2
2	C2	2
3	W	1
4	no. of particles	25 (for lower dimensions)
5	no. particles	50 (higher dimensions)
For TA part		
1	eps	0.01
2	acc	0.0000012
3	thrtol	0.000001
3	pindex	29
4	limit	50
5	gitr	25

Table 2: Details of Test functions

SNo.	Name of the function	Dimension	Reported Objective function value
1	Sphere (Sphere, 2010)	2,30	0
2	RosenBrock (Schwefel,1995, More,1981)	2,20	0
3	Goldstein (Dixon 1978)	2	3
4	Schaffer1 (Michael 1996)	2	0
5	Schaffer2 (Michael 1996)	2	0
6	Bohachevsky1(Bohachevsky, 1986)	2	0
7	Bohachevsky2 (Bohachevsky 1986)	2	0
8	Periodic (Price 1977)	2	0.9
9	Camelback6 (Dixon 1978, Michael 1996)	2	0
10	Becker and Lago (Price 1977)	2	0
11	Ackley (Storn 1997)	2	0
12	Salomon (Salomon 1995)	5	0
13	Kowalik (Jansson 1995)	4	3.0748 * 10-4
14	Levy and montalov1 (Levy 1985)	3,30	0
15	Levy and montalov2 (Levy 1985, Dekker 1991)	5,30	0
16	Meyer and Roth (Wolfe 1978)	3	0
17	Miele and Cantrell (Wolfe 1978)	4	0
18	Neumaier2 (Neumaier 2010)	4	0
19	Powells (Wolfe 1978)	4	0
20	Woods (Michal 1996, Wolfe 1978)	4	0
21	Zakharov (Zakharov, 2010)	2,20,30,50	0
22	Axis Parallel Hyper Ellipsoidal (Axis 1010)	2,30	0
23	Rastrigin (Storn 1997, Torn 1989)	2	0
24	Shekel5 (Dixon 1978)	4	-10.1499
25	Shekel7 (Dixon 1978)	4	-10.3999
26	Shekel10 (Dixon 1978)	4	-10.5319

Table 3: Objective function values and functional evaluations for lower dimension problems

Function number	**PSO**	**FE's of PSO**	**PSOTA**	**FE's of PSOTA**
1	0.000000	1275	0.000000	830
2	0.000000	10025	0.000003	7401
3	3.000000	1275	3.000000	1299
4	0.003715	14040	0.004210	13780
5	0.000000	6275	0.000000	5024
6	0.000000	14040	0.000000	11844
7	0.000000	14040	0.000000	11735
8	0.900000	7525	0.900000	7316
9	0.000000	2275	0.000000	957
10	0.000000	2025	0.000000	1171
11	0.009334	3775	0.000004	1957
12	0.099833	17525	0.006845	14029
13	0.000307	12525	0.000308	4407
14	0.000000	3775	0.000000	1218
15	0.000000	3775	0.000000	3039
16	0.000083	12525	0.000069	10386
17	0.000005	5025	0.000003	2591
18	0.010448	15025	0.007210	13005
19	0.000001	5025	0.000005	3509
20	0.020200	25025	0.029904	18204
21	0.000000	1275	0.000000	911
22	0.000000	650	0.000003	388
23	0.000000	3775	0.000000	2624
24	-9.309608	162525	-10.153199	75109
25	-10.035613	162525	-10.402944	77128
26	-9.997189	162535	-10.536411	77681

Table 4: Objective function values and functional values for higher dimensional problems

Function number	**PSO**	**FE's of PSO**	**PSOTA**	**FE's of PSOTA**
1 @ 30d	0.000005	25030	0.000000	12292
14@ 30d	0.017285	35050	0.000000	24125
15@ 30d	0.000000	75050	0.000000	33178
21@ 20d	0.000000	20040	0.000003	15374
21@ 30d	0.000000	35050	0.000000	30050
21@ 50d	0.000724	75050	0.000054	58956
22@ 30d	0.000000	45050	0.000001	26766
2@ 20d	14.854886	400050	0.069339	400100

Table 5: Success Rate (SR) and Standard Deviation (S.D.) for lower dimensional problems

Function number	SR % of PSO	S.D. of PSO	SR% of PSOTA	S.D. of PSOTA
1	100	0	100	9.68468E-07
2	100	0	93.33	9.63805E-06
3	100	0	100	8.89918E-07
4	57	0.004706	57	0.004897
5	96.7	1.82574E-07	96.7	7.48147E-06
6	100	0	100	0
7	100	0	100	0
8	100	4.51681E-16	100	4.51681E-16
9	100	1.82574E-07	100	2.53708E-07
10	100	1.82574E-07	100	1.11211E-06
11	96.66	0.051124	93.33	1.29E-05
12	0	2.82E-17	83.33	0.025298
13	100	1.82574E-07	90	1.17248E-06
14	100	4.61133E-07	100	4.61133E-07
15	100	0	100	0
16	23.33	0.000114927	23.33	2.90144E-05
17	96	3.32169E-05	100	1.19998E-05
18	0	0.013178	3.33	0.009086
19	80	1.02889E-06	80	5.46893E-06
20	0	0.043618	0	0.023845
21	100	1.82574E-07	100	1.82574E-07
22	96.66	0	96.7	5.16E-06
23	100	0	100	0
24	83	1.918587	100	1.81E-15
25	90	1.336582506	100	4.49776E-07
26	90	1.645332	100	1.81E-15

Table 6: Success Rate (SR) and Standard Deviation (S.D.) for higher dimensional problems

Function number	SR% of PSO	S.D. of PSO	SR% of PSOTA	S.D. PSOTA
1 @ 30d	100	1.64701E-06	100	0
14@ 30d	83.33	0.039312	97	1.83077E-06
15@ 30d	96.6	0.004159	100	1.82574E-07
21@ 20d	100	0	96.67	8.13966E-06
21@ 30d	96.6	3.83705E-06	100	6.68675E-07
21@ 50d	23.33	0.001633	33.33	8.79E-05
22@ 30d	100	0	100	2.34128E-06

Table 7: Parameter Settings for Different Test Functions

Function	N	hms	hmcr	par	fw	p	up	diff1	diff2
Aluffi-Pentini	2	30	0.1	0.2	0.5	5	0.7	0.01	0.00001
Becker	2	20	0.1	0.2	0.002	5	0.3	0.000001	0.000001
Bohachevsky1	2	50	0.1	0.5	0.5	9	0.4	0.000001	0.000001
Bohachevsky2	2	50	0.1	0.5	0.5	9	0.4	0.0001	0.000001
Camelback3,6	2	20	0.1	0.2	0.35	9	0.3	0.0001	0.00001
Dekkers	2	20	0.1	0.8	0.35	9	0.4	0.0001	0.000001
Easom	2	40	0.1	0.6	0.025	5	0.8	0.000001	0.000001
Goldstein	2	20	0.1	0.75	0.25	9	0.4	0.000001	0.000001
Hartman 3	3	20	0.1	0.75	0.002	9	0.4	0.0001	0.000001
Miele	4	20	0.1	0.4	0.005	9	0.4	0.0001	0.000001
Mod.Rosenbrock	2	20	0.1	0.2	0.003	9	0.4	0.000001	0.000001
Paviani	1	20	0.1	0.7	0.35	9	0.4	0.0001	0.000001
Periodic	2	20	0.1	0.2	0.35	9	0.4	0.000001	0.000001
Powell	4	30	0.1	0.5	23	9	0.4	0.00001	0.000001
Rosenbrock	2	20	0.1	0.2	0.25	9	0.3	0.0001	0.00001
Salomon	1	20	0.1	0.2	0.002	9	0.4	0.000001	0.000001
Schaffer1,2	2	30	0.1	1	0.24	9	0.4	0.000001	0.000001
Schwefel	5	20	0.1	0.8	0.35	9	0.4	0.000001	0.000001
Sphere	5	20	0.1	0.7	0.3	9	0.4	0.0001	0.000001
Zakharov	5	30	0.1	0.8	0.25	9	0.4	0.001	0.000001
Rosenbrock(20 variables)	20	110	0.1	0.9	0.24	9	0.4	0.000001	0.000001
Zakharov (20 variables)	20	110	0.1	0.8	0.23	9	0.4	0.000001	0.000001

Table 8: The Best, Worst and Average Results Based On the 30 Simulation Runs

		MHS				MHS+MGDA			
SNo	**FUNCTION**	**BEST**	**WORST**	**AVG**	**F.EVAL**	**BEST**	**WORST**	**AVG**	**F.EVAL**
1	Aluffi Penttini	-0.3523	-0.349582	-0.352032	4497	-0.35238	-0.351349	-0.35222	3015
2	Becker	0.00001	0.000202	0.000013	6582	0	0.00245	0.000261	2201
3	Bohachevsky1	0	0.08717	0.009631	6619	0	0.0546	0.0116	4162
4	Bohachevsky2	0	0.188002	0.011379	7165	0	0.023859	0.006072	4032
5	Camelback3	0.000002	0.001106	0.000263	4261	0	0.000614	0.000317	4081
6	Camelback6	-1.0316	-1.029675	-1.031211	4478	-1.03159	-1.027918	-1.03113	4243
7	Dekker's	-24776.515	-24554.314	-24756.583	3444	-24776.513	-24510.773	-24772.369	2276
8	Easom	-0.9665	0	-0.966595	5543	-0.99999	0	-0.96656	4038
9	Goldstein	3.00016	3.075531	3.01732	3888	3.00057	3.424754	3.060314	1033
10	Hartman3	-3.8627	-3.862445	-3.862748	5840	-3.86278	-3.862776	-3.86278	3661
11	Miele	0	0	0	7717	0.000001	0.000088	0.000018	2853
12	Mod.Rosenbro-ck	0	0.139236	0.020783	7353	0	0.131673	0.0267	4570
13	Paviani	3.666673	3.666685	3.666673	1364	3.666673	3.666741	3.666682	1552
14	Periodic	0.900002	0.902353	0.900265	7015	0.900005	0.913871	0.90078	4480
15	Powell	0.000811	1.242686	0.142005	10039	0.000048	1.107911	0.0134726	6687
16	Rosenbrock	0.00046	0.210528	0.044408	6597	0.000001	0.052244	0.004537	6662
17	Salomon	0	0.177213	0.042011	1480	0.00017	0.200012	0.05187	1042
18	Schaffer2	0.224512	3.488596	0.710002	4234	0.036513	0.500399	0.0172575	3678
19	Schaffer1	0.009716	0.039059	0.014096	4442	0.00917	0.037235	0.012822	3107
20	Schwefel	-2094.9	-2090.9270	-2094.040	8755	-2094.91	-2094.914	-2094.91	8726
21	Sphere	0.005837	0.039084	0.005837	8577	0.00019	0.002044	0.001032	4771
22	Zakharov	0.00019	0.103108	0.01826	9468	0.00036	0.003827	0.0015	5888
23	RosenBrock (20 variables)	0	0.605928	0.085163	9875	0.000006	0.487838	0.161195	7921
24	Zakharov (20 iables)	0.00393	0.082719	0.033866	10240	0.000100	0.002578	0.000750	10014

Table 9: Comparison of average accuracies and functional evaluations on benchmark functions

		MHS		MHS+MGDA		MHS+TA	
SNo	**FUNCTION**	**AVG**	**FEval**	**AVG**	**FEval**	**AVG**	**FEval**
1	Aluffi Penttini	-0.352032	4497	-0.352226	3015	-0.352343	1408
2	Becker	0.000013	6582	0.000261	2201	0.000027	1456
3	Bohachevsky1	0.009631	6619	0.0116	4162	0.000001	1883
4	Bohachevsky2	0.011379	7165	0.006072	4032	0.000001	1883
5	Camelback3	0.000263	4261	0.000317	4081	0.000050	1574
6	Camelback6	-1.031211	4478	-1.03113	4243	-1.031431	1336
7	Dekker's	-24756.5839	3444	-24772.369	2276	-24771.97851	1513
8	Easom	-0.966595	5543	-0.96656	4038	-0.998774	1653
9	Goldstein	3.01732	3888	3.060314	1033	3.002347	1432
10	Hartman3	-3.862748	5840	-3.86278	3661	-3.862746	1771
11	Miele	0	7717	0.000018	2853	0.000025	1770
12	Mod.Rosenbrock	0.020783	7353	0.0267	4570	0.005817	2104
13	Periodic	0.900265	7015	0.90078	4480	0.906822	1869
14	Powell	0.142005	10039	0.0134726	6687	0.022234	4952
15	Salomon 10 d	0.042011	1480	0.05187	1042	0	1324
16	Schaffer2	0.710002	4234	0.0172575	3678	0.355321	1602
17	Schaffer1	0.014096	4442	0.012822	3107	0.012209	2217
18	Schwefel (10 d)	-2094.040	8755	-2094.914	8726	-4187.6411	10039
19	Sphere (5d)	0.005837	8577	0.001032	4771	0.000133	4037
20	Zakharov	0.01826	9468	0.0015	5888	0.000215	5478

Table 10: Average Accuracies of ten folds obtained on different datasets

Dataset (no. features)	Hybrid model 1(TALR)	Hybrid model 2 (MLP)
	Classification accuracy	**Classification accuracy**
Spanish bank (9)	98.33	57.5
Turkish bank (12)	100	67.5
UK bankruptcy (10)	84.94	49.9
UK credit (11)	76.96	73.63
Dotcom companies (24)	79.16	72.49
Pima Indians (8)	71.833	64.89
Spectf (44)	84.2	80.59

Table 11: Features selected by the Hybrid models

Sno	Dataset	A	B	A U B
1	Spanish bank (9)	F2	F5,F6,F7,F9	F2,F5,F6,F7,F9
2	Turkish bank (12)	F5,F6,F11	F2,F8,F10,F11	F2,F5,F6,F8,F10,F11
3	UK bankruptcy (10)	F6,F10	F2,F4,F5,F9,F10	F2,F4,F5,F6,F9,F10
4	UK credit (11)	F1,F2,F4,F5,F6	F4,F9	F1,F2,F4,F5,F6,F9
5	Dotcom companies (24)	F1,F3,F10,F16, F20,F21,F22, F24	F7,F11,F12,F17,F21	F1,F2,F3,F7,F10,F11, F12,F16,F17,F20,F21, F22,F24
6	Pima Indians (8)	F3,F5	F2,F3,F4,F6	F2,F3,F4,F5,F6
7	Spectf (44)	F1,F3,F14,F24, F33	F1,F2,F3,F4,F5,F7, F8,F11,F14,F17,F18, F20,F21,F22,F24,F27, F31,F32,F33,F34, F35,F37	F1,F2,F3,F4,F5,F7,F8, F11,F14,F17,F18,F20, F21,F22,F24,F27,F31, F32,F33,F34,F35,F37

Table 12: Financial ratios of Spanish banks

Features	Predictor Variable Name	
F1	*Current Assets/Total Assets*	*CA/TA*
F2	*Current Assets-Cash/Total Assets*	*CAC/TA*
F3	*Current Assets/Loans*	*CA/L*
F4	*Reserves/Loans*	*R/L*
F5	*Net Income/Total Assets*	*NI/TA*
F6	*Net Income/Total Equity Capital*	*NI/TEC*
F7	*Net Income/Loans*	*NI/L*
F8	*Cost Of Sales/Sales*	*CS/S*
F9	*Cash Flow/Loans*	*CF/L*

Table 13: Financial ratios of Turkish bank

Features	Predictor Variable Name
F1	*Interest Expenses/ Average Profitable Assets*
F2	*Interest Expenses/ Average non-Profitable Assets*
F3	*(Share holders' Equity + Total income)/(Deposits + Non-deposit funds)*
F4	*Interest Income / Interest expenses*
F5	*(Share holders' Equity + Total income)/ Total assets*
F6	*(Share holders' Equity + Total income)/(Total assets + Contingencies & commitments)*
F7	*Networking capital / Total assets*
F8	*(Salary and Employees' benefits + Reserve for retirement)/ No. of personnel*
F9	*Liquid Assets / (Deposits + non-deposit funds)*
F10	*Interest expenses / Total expenses*
F11	*Liquid assets / total assets*
F12	*Standard Capital ratio*

Table 14: Financial ratios of UK Credit dataset

Features	Predictor variable name	
F1	*Year of birth*	*dob*
F2	*Number of children*	*nkid*
F3	*Number of other dependents*	*dep*
F4	*Spouse's income*	*sinc*
F5	*Applicant's income*	*dainc*
F6	*Value of Home*	*dhval*
F7	*Mortgage balance outstanding*	*Dmort*
F8	*Outgoings on mortgage or rent*	*doutm*
F9	*Outgoings on Loans*	*doutl*
F10	*Outgoings on Hire Purchase*	*douthp*
F11	*Outgoings on credit cards*	*doutcc*

Table 15: Financial ratios of UK Bankruptcy

Features	Predictor Variable Name	
F1	*Sales*	*Sales*
F2	*Profit Before Tax/Capital Employed (%)*	*PBT/CE*
F3	*Funds Flow/Total Liabilities*	*FF/TL*
F4	*(Current Liabilities + Long Term Debits)/Total Assets*	*(CL+LTD)/TA*
F5	*Current Liabilities/Total Assets*	*CL/TA*
F6	*Current Assets/Current Liabilities*	*CA/CL*
F7	*Current Assets-Stock/Current Liabilities*	*CA-S/CL*
F8	*Current Assets-Current Liabilities/Total Assets*	*CA-CL/TA*
F9	*LAG(Number of days between account year end and the date of annual report)*	*LAG*
F10	*Age*	*Age*

Table 17: Features of Pima Indians dataset

Features	Predictor Variable Name
F1	*Number of times pregnant*
F2	*Plasma glucose concentration a 2 hours in an oral glucose tolerance test*
F3	*Diastolic blood pressure (mm Hg)*
F4	*Triceps skin fold thickness (mm)*
F5	*2-Hour serum insulin (mu U/ml)*
F6	*Body mass index (weight in kg/(height in m)^2)*
F7	*Diabetes pedigree function*
F8	*Age (years)*

Table 16: Features of Dotcom dataset

Features	Predictor Variable Name	Financial ratios
F1	*Working capital/total assets*	*WC/TA*
F2	*Total debt/Total assets*	*TD/TA*
F3	*Current assets/current liabilities*	*CA/CL*
F4	*Operating income/total assets*	*OI/TA*
F5	*Net income/total assets*	*NI/TA*
F6	*Cash flow/total debt*	*CF/TD*
F7	*Quick assets/current liabilities*	*QA/CL*
F8	*Cash flow/sales*	*CF/S*
F9	*Retained earnings/total assets*	*RE/TA*
F10	*Sales/total assets*	*S/TA*
F11	*Gross profit/total assets*	*GP/TA*
F12	*Net income/shareholders equity*	*NI/SE*
F13	*Cash/total assets*	*C/TA*
F14	*Inventory/sales*	*I/S*
F15	*Quick assets/total assets*	*QA/TA*
F16	*Price per share/earnings per share*	*P/E*
F17	*Sales/market capitalization*	*S/MC*
F18	*Current assets/total assets*	*CA/TA*
F19	*Long term debt/total assets*	*LTD/TA*
F20	*Operating income/sales*	*OI/S*
F21	*Operating income/market capitalization*	*OI/MC*
F22	*Cash/sales*	*C/S*
F23	*Current assets/sales*	*CA/S*
F24	*Net income/(total assets-total liabilities)*	*NI/(TA-TL)*

Table 18: Features of *Spectf* dataset

Features	Predictor Variable name
F1	*F1R: continuous (count in ROI (region of interest) 1 in rest)*
F2	*F1S: continuous (count in ROI 1 in stress)*
F3	*F2R: continuous (count in ROI 2 in rest)*
F4	*F2S: continuous (count in ROI 2 in stress)*
F5	*F3R: continuous (count in ROI 3 in rest)*
F6	*F3S: continuous (count in ROI 3 in stress)*
F7	*F4R: continuous (count in ROI 4 in rest)*
F8	*F4S: continuous (count in ROI 4 in stress)*
F9	*F5R: continuous (count in ROI 5 in rest)*
F10	*F5S: continuous (count in ROI 5 in stress)*
F11	*F6R: continuous (count in ROI 6 in rest)*
F12	*F6S: continuous (count in ROI 6 in stress)*
F13	*F7R: continuous (count in ROI 7 in rest)*
F14	*F7S: continuous (count in ROI 7 in stress)*
F15	*F8R: continuous (count in ROI 8 in rest)*
F16	*F8S: continuous (count in ROI 8 in stress)*
F17	*F9R: continuous (count in ROI 9 in rest)*
F18	*F9S: continuous (count in ROI 9 in stress)*
F19	*F10R: continuous (count in ROI 10 in rest)*
F20	*F10S: continuous (count in ROI 10 in stress)*
F21	*F11R: continuous (count in ROI 11 in rest)*
F22	*F11S: continuous (count in ROI 11 in stress)*
F23	*F12R: continuous (count in ROI 12 in rest)*
F24	*F12S: continuous (count in ROI 12 in stress)*
F25	*F13R: continuous (count in ROI 13 in rest)*
F26	*F13S: continuous (count in ROI 13 in stress)*
F27	*F14R: continuous (count in ROI 14 in rest)*
F28	*F14S: continuous (count in ROI 14 in stress)*
F29	*F15R: continuous (count in ROI 15 in rest)*
F30	*F15S: continuous (count in ROI 15 in stress)*
F31	*F16R: continuous (count in ROI 16 in rest)*
F32	*F16S: continuous (count in ROI 16 in stress)*
F33	*F17R: continuous (count in ROI 17 in rest)*
F34	*F17S: continuous (count in ROI 17 in stress)*
F35	*F18R: continuous (count in ROI 18 in rest)*
F36	*F18S: continuous (count in ROI 18 in stress)*
F37	*F19R: continuous (count in ROI 19 in rest)*
F38	*F19S: continuous (count in ROI 19 in stress)*
F39	*F20R: continuous (count in ROI 20 in rest)*

F40	*F20S: continuous (count in ROI 20 in stress)*
F41	*F21R: continuous (count in ROI 21 in rest)*
F42	*F21S: continuous (count in ROI 21 in stress)*
F43	*F22R: continuous (count in ROI 22 in rest)*
F44	*F22S: continuous (count in ROI 22 in stress)*

Figures

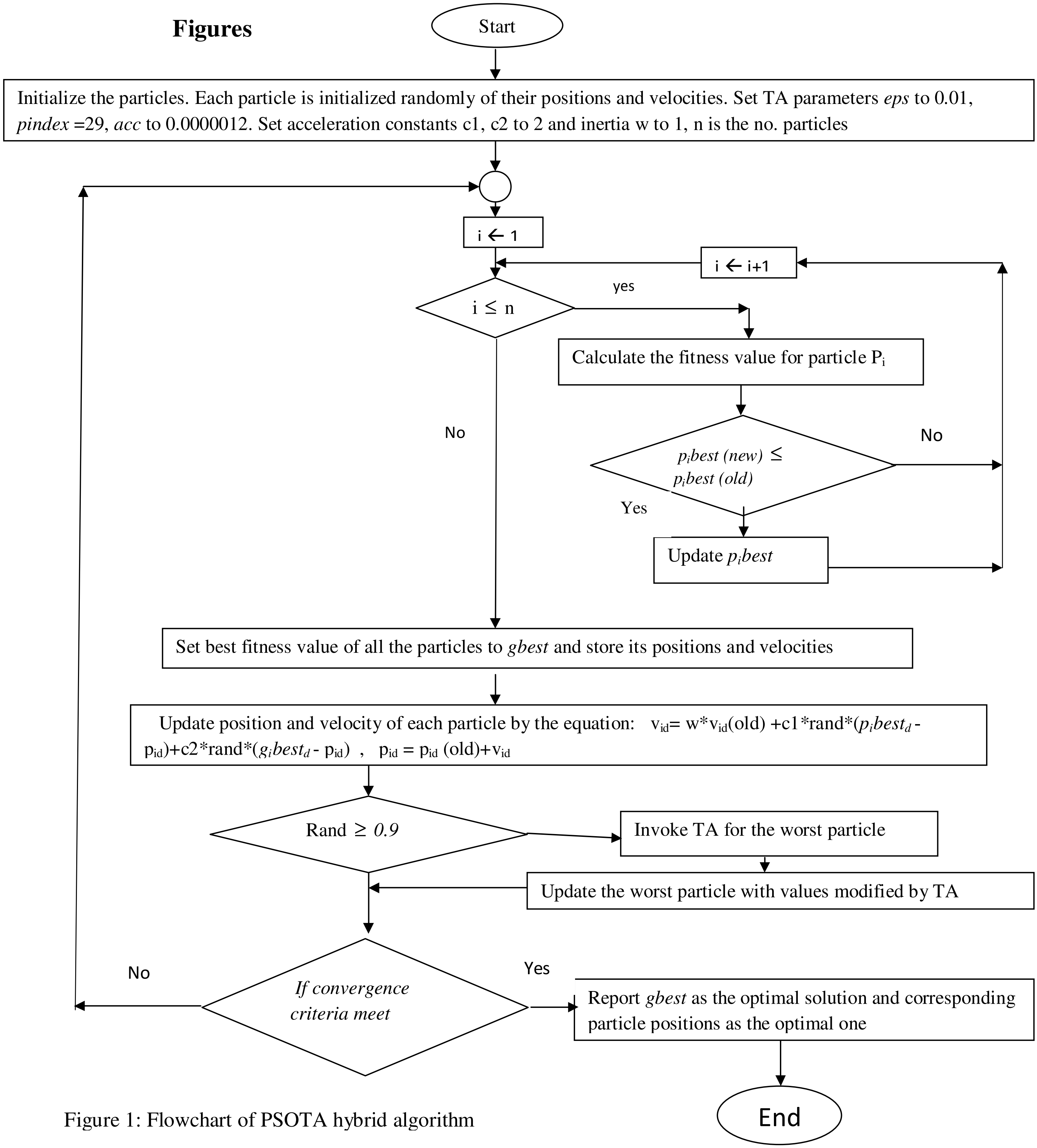


Figure 1: Flowchart of PSOTA hybrid algorithm

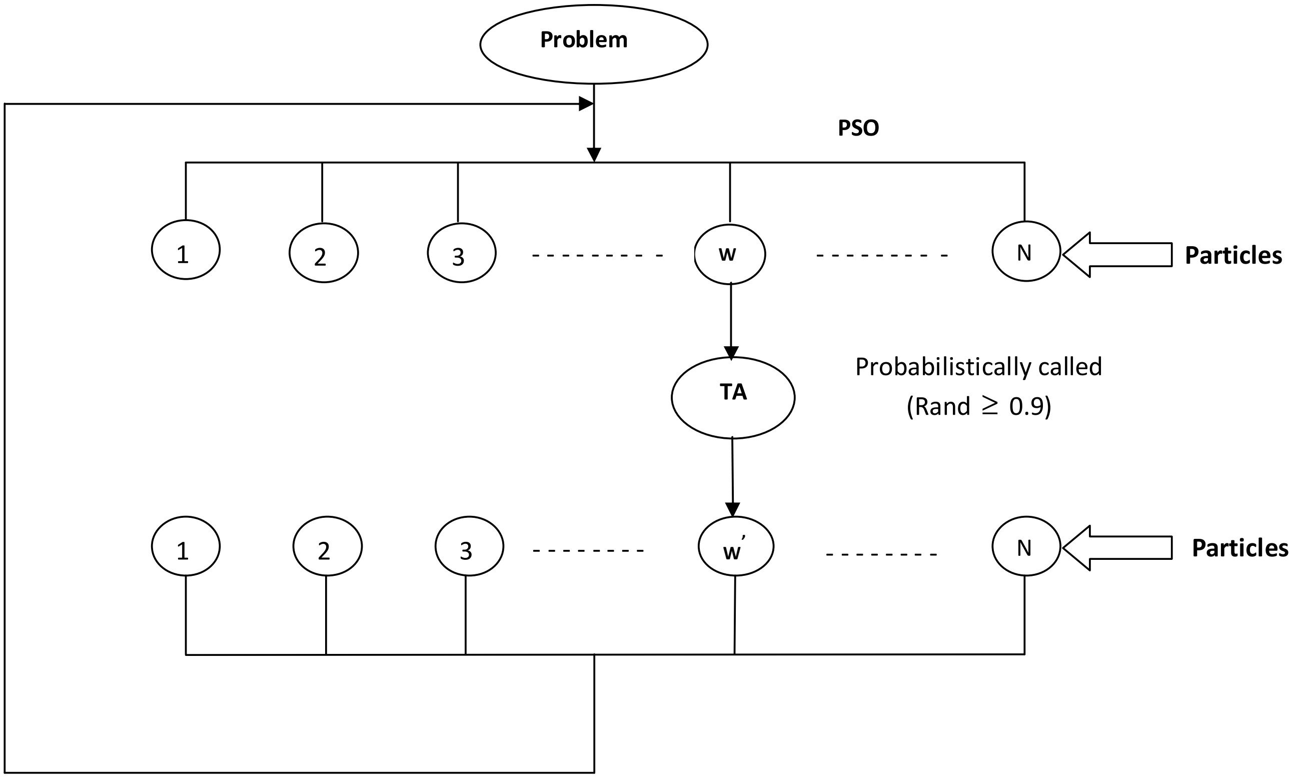


Figure 2: Schematic representation of our proposed hybrid: W represents the worst particle, N is the number of particles and Rand represents random number generator.

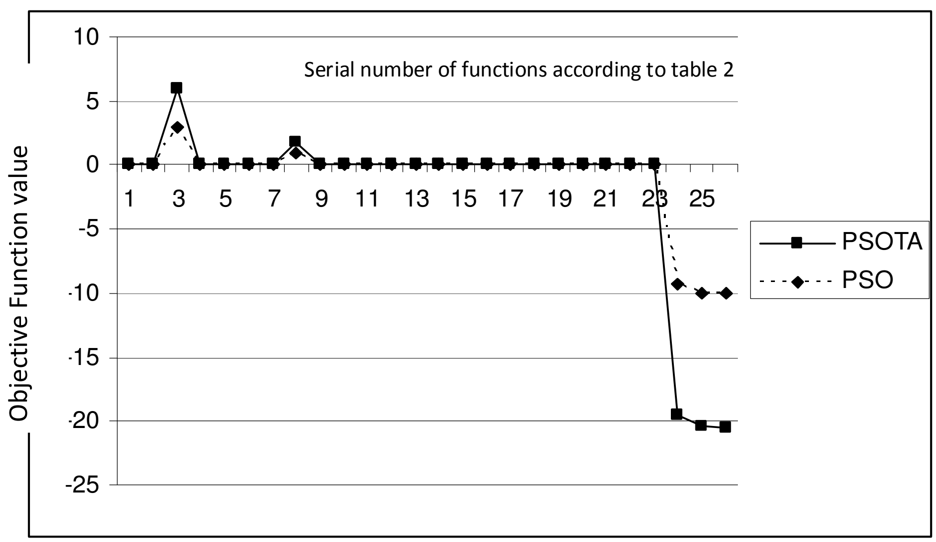


Figure 3: Graph comparing accuracy of fitness values of PSO and PSOTA for lower dimensions

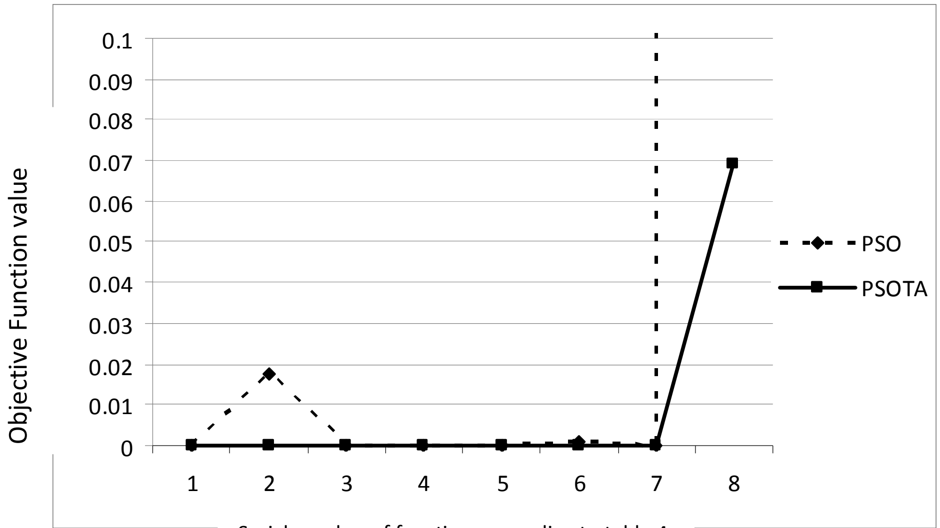


Figure 4: Graph comparing accuracy of fitness values of PSO and PSOTA for higher dimensions

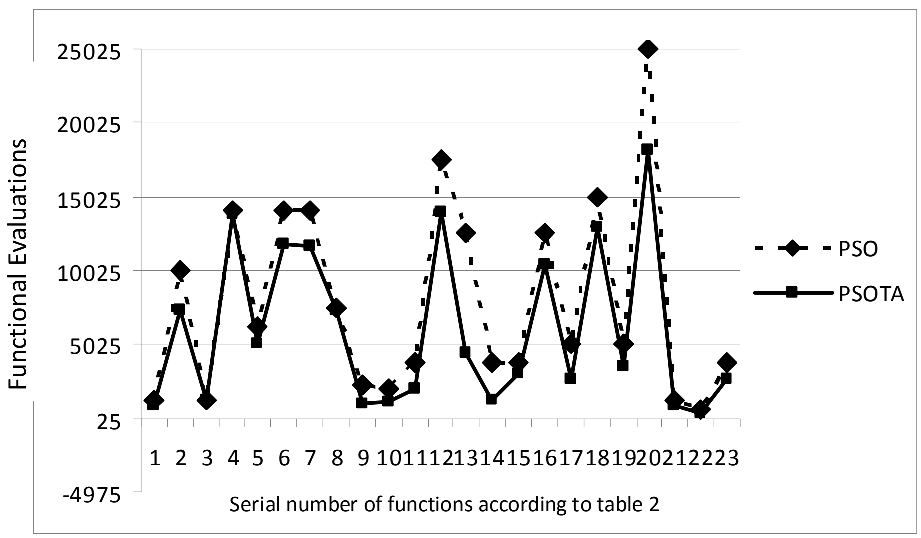


Figure 5: Graph comparing accuracy of functional evaluations of PSO and PSOTA for lower dimensions

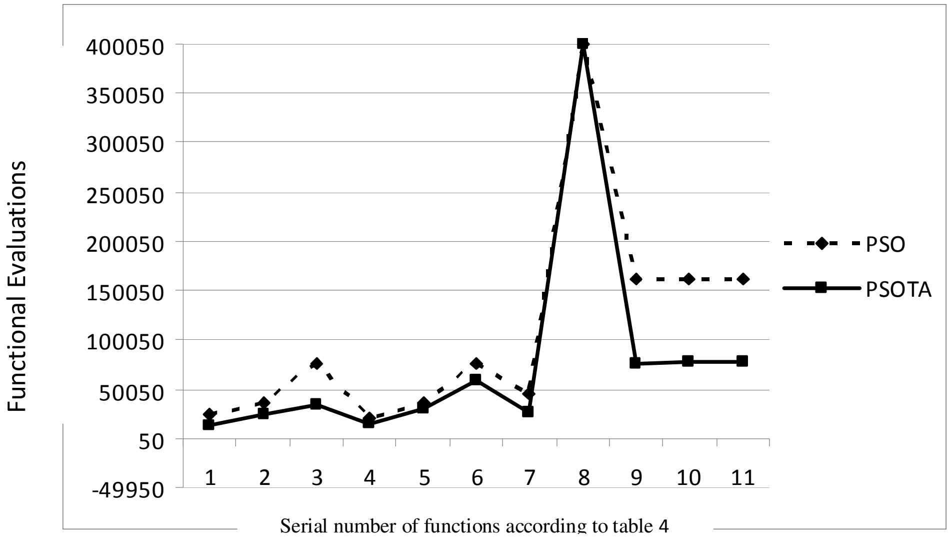
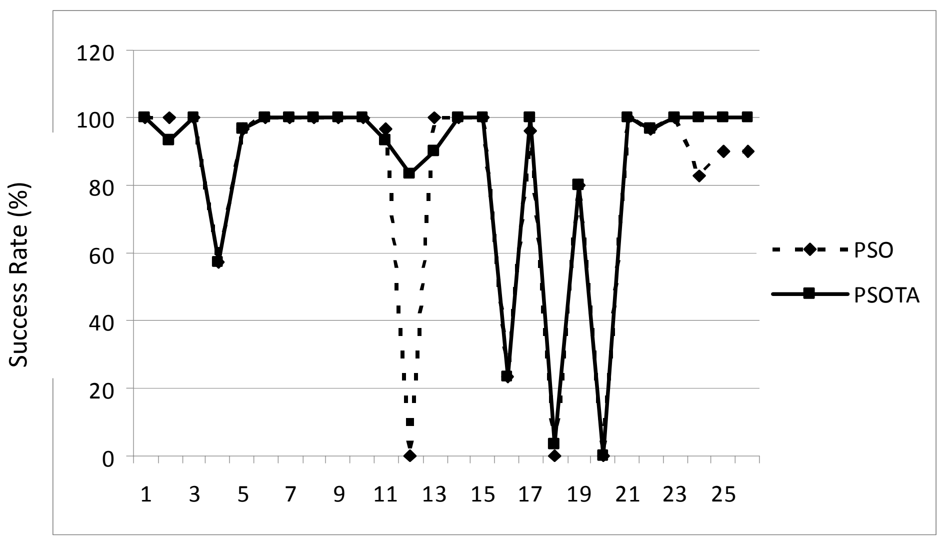


Figure 6: Graph comparing accuracy of functional evaluations of PSO and PSOTA for higher dimensions



Serial number of functions according to table 2

Figure 7: Graph comparing accuracy of success rates of PSO and PSOTA for lower dimensions

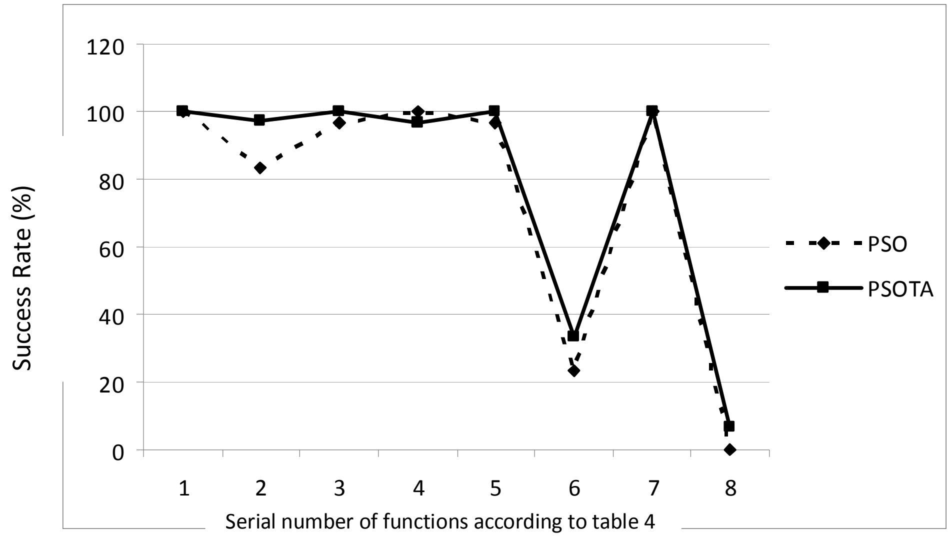


Figure 8: Graph comparing accuracy of success rates of PSO and PSOTA for higher dimensions

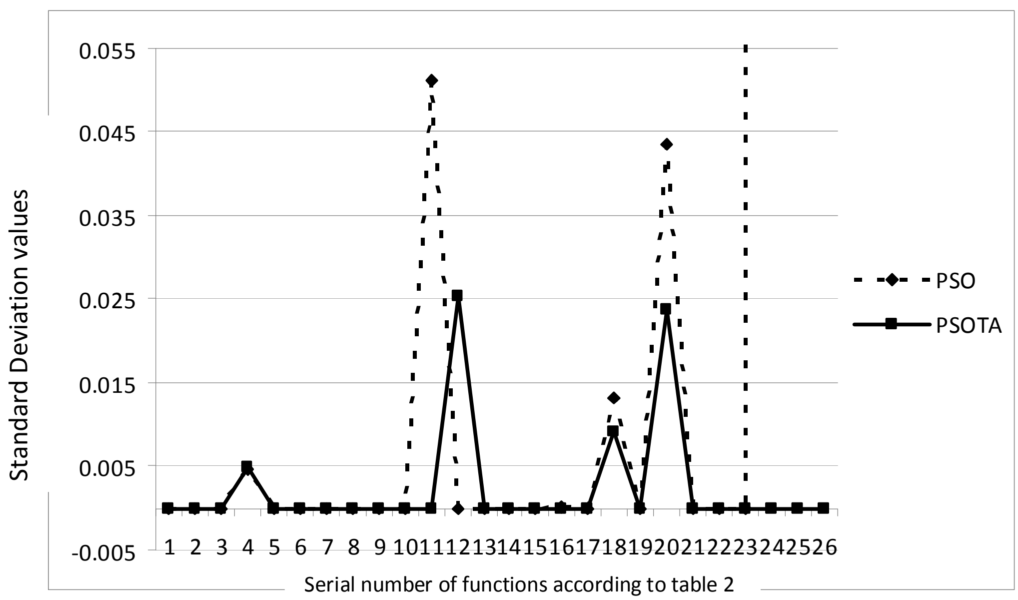


Figure 9: Graph comparing accuracy of standard deviations of PSO and PSOTA for lower dimensions

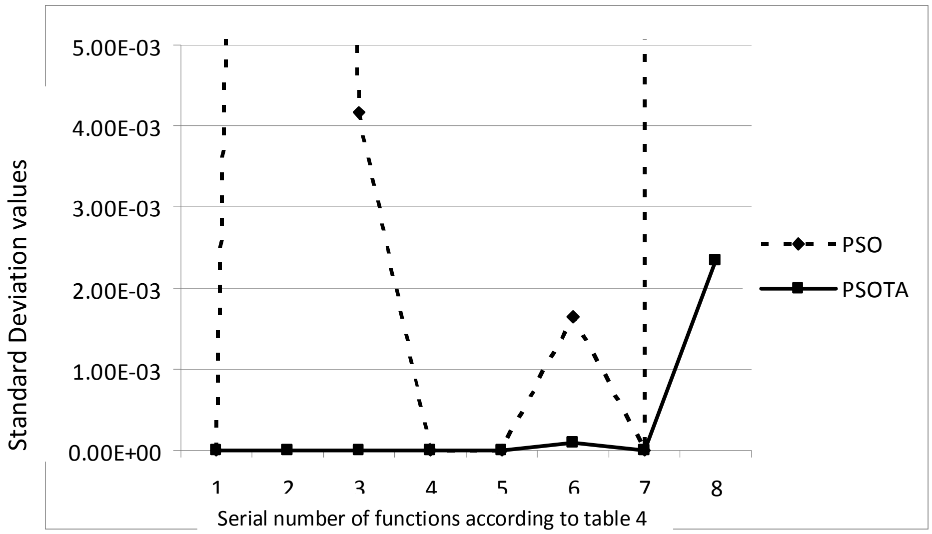


Figure 10: Graph comparing accuracy of fitness values of PSO and PSOTA for higher dimensions

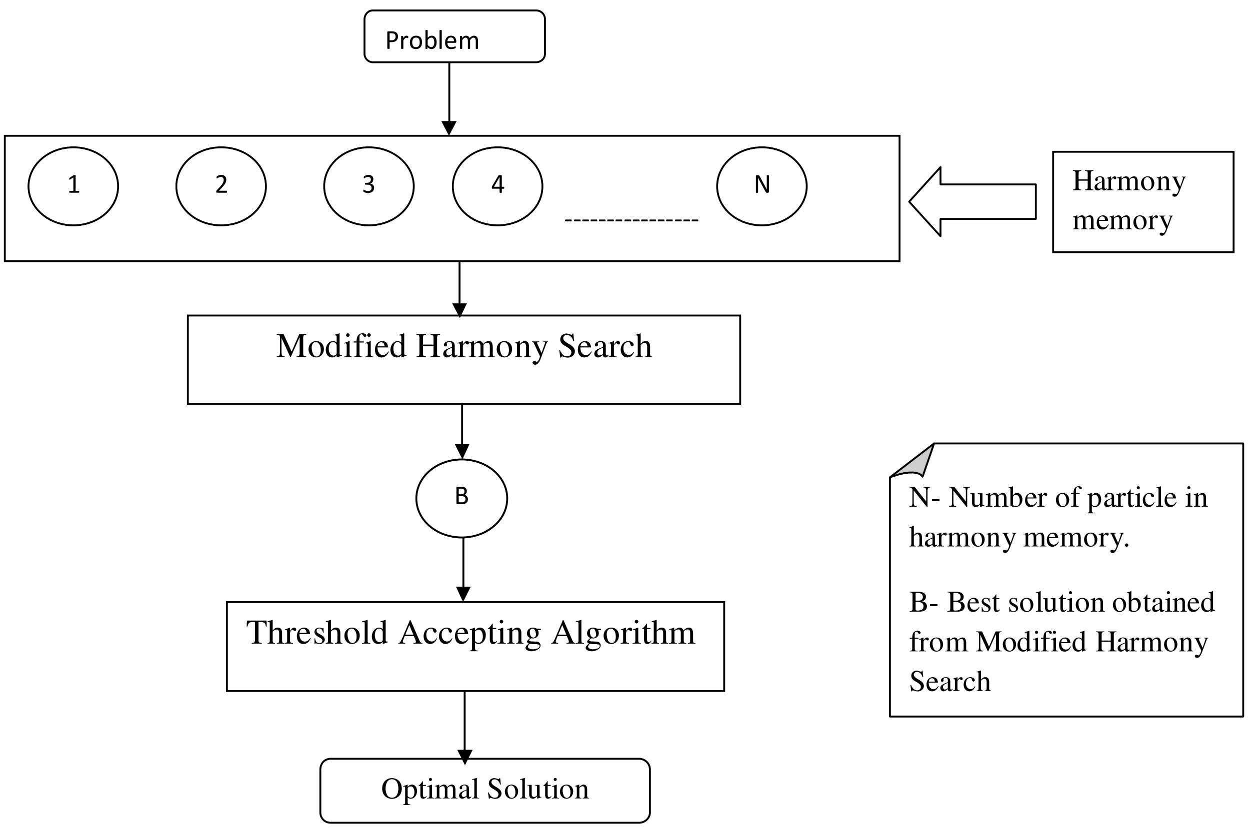


Figure 11: Schematic view of MHS+TA model

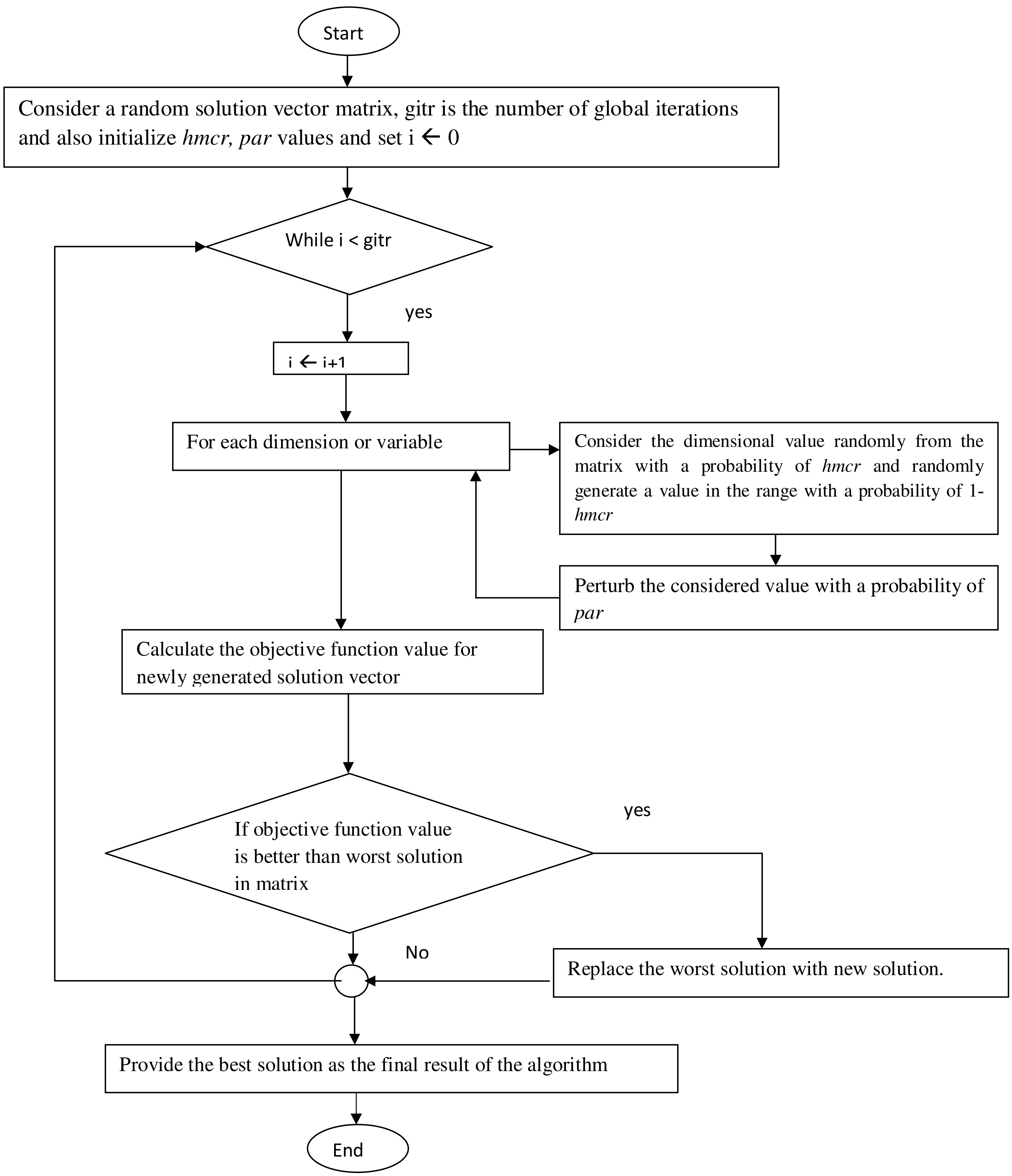


Figure 12: Flowchart of Harmony Search

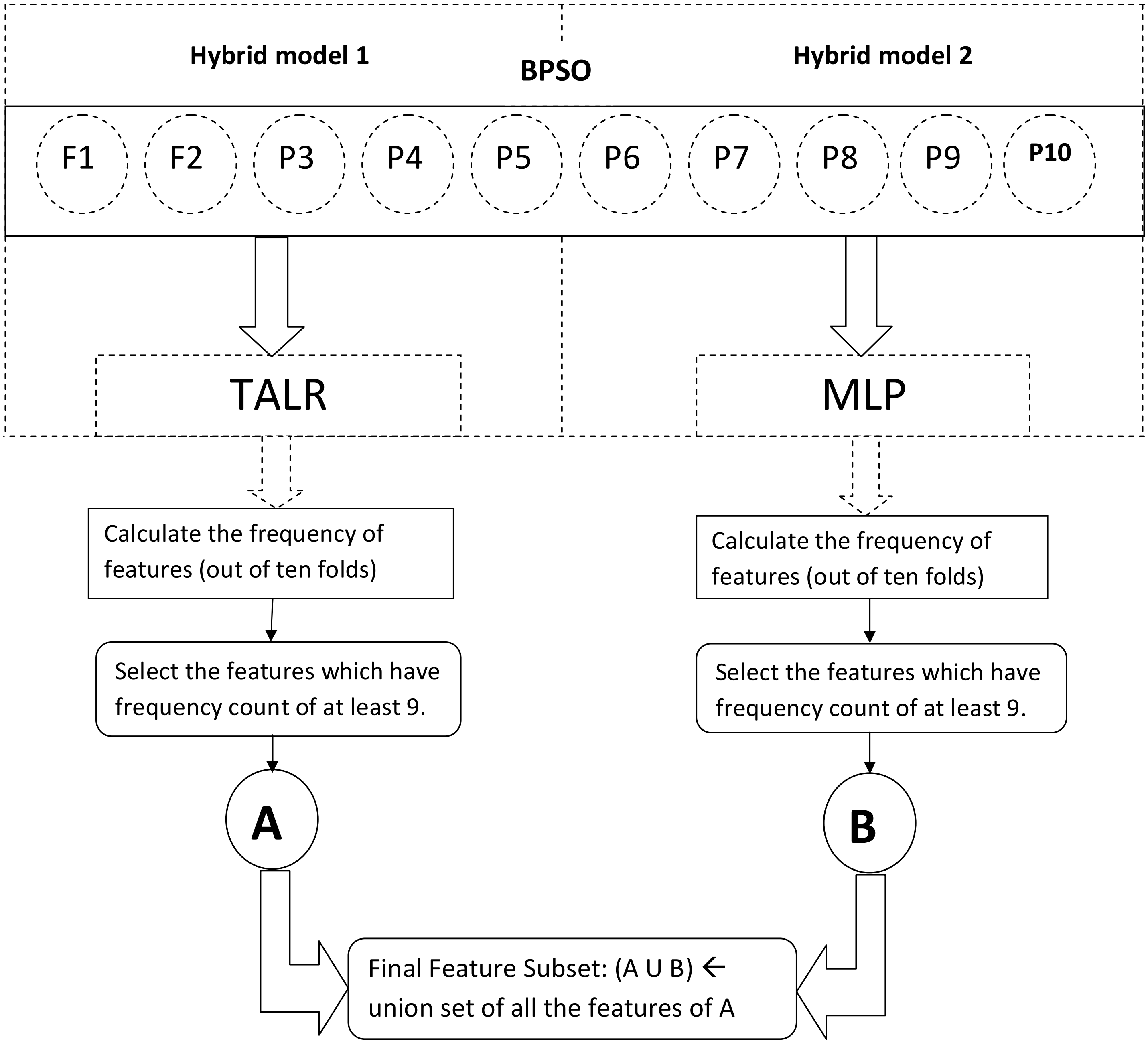


Figure 13: Architecture of our Proposed Model